1994

# Generalized FFTS - A Survey of Some Recent Results

David K. Maslen

Daniel N. Rockmore
*Dartmouth College*

# GENERALIZED FFTS - A SURVEY OF SOME RECENT RESULTS

DAVID K. MASLEN AND DANIEL N. ROCKMORE

Abstract. In this paper we survey some recent work directed towards generalizing the fast Fourier transform (FFT). We work primarily from the point of view of group representation theory. In this setting the classical FFT can be viewed as a family of efficient algorithms for computing the Fourier transform of either a function defined on a finite abelian group, or a bandlimited function on a compact abelian group. We discuss generalizations of the FFT to arbitrary finite groups and compact Lie groups.

## 1. Background

The classical Fast Fourier Transform (FFT) is one of the most useful algorithms ever developed. It is the cornerstone of many digital signal processing algorithms and as such impacts our lives daily. The best known of these algorithms is the Cooley-Tukey FFT. Originally discovered by Gauss, as an efficient means of interpolating asteroid orbits [41] and later rediscovered by Cooley and Tukey for the efficient analysis of time series [26] this algorithm permits the fast and reliable computation of the Discrete Fourier Transform (DFT), which gives the decomposition of a periodic function into a linear combination of sines and cosines. For historical discussions of the development see [24, 25, 51].

There are various contexts in which this algorithm and its many variants may be formulated. Our point of view is a representation theoretic one. For us, an FFT is an algorithm which gives an efficient decomposition of a function on a group, or its coset space, into a sum of irreducible matrix coefficients. In this setting the classical FFT is an algorithm for abelian groups and it is natural to look for generalizations of these techniques to other groups.

The first such results in this direction appear to be due to Willsky [92] who was looking for new algorithms for filter design. Others also did much to lay the groundwork in this relatively new interdisciplinary area looking to extend the uses of the FFT in signal processing [57, 11, 22] and data analysis [31]. In this paper we will survey some of the recent results about such generalizations. We focus primarily upon recent work using factorization of group elements, in the finite group case, and the analogous approach for compact groups [68, 69, 64]. We shall consider the algorithmic aspects of this work and leave a discussion of the potential applications for another paper in this volume [80].

We proceed as follows. In Section 2 we discuss two abelian FFTs, an FFT due to Yates, as well as the Cooley-Tukey FFT. These algorithms may be given formulations that reveal many properties of generalized FFTs. They are based on

1991 *Mathematics Subject Classification.* 20C40; Secondary 65T10, 42C10.
Max-Planck-Institut für Mathematik.
NSF DMS 9404275, 9304580, and PFF, AFOSR DOD F4960-93-1-0567.

the factorization of group elements, the factorization of representations as tensor products, and the use of bases well-behaved under the restriction of representations to a subgroup. An important feature of our approach to Cooley-Tukey is a scheme for indexing basis vectors in the representations by paths in a related diagram.

Section 3 discusses the relevant generalization of the Fourier transform to finite groups. The majority of this section is devoted to a summary of the "separation of variables" methodology which is based on the use of adapted bases and factorization of group elements. This is a completely general technique for constructing FFTs for finite groups and besides yielding many new FFT algorithms, it also gives a framework for deriving many earlier FFTs, both abelian and nonabelian, in a uniform fashion. We also consider the problem of how to find the best FFT that uses these techniques. Separation of variables is one approach for FFT design. We explain others at the end of the section.

In Section 4 we turn to the continuous compact case. Here we pursue the point of view that explains the Cooley-Tukey FFT as an efficient expansion of a band-limited function on the circle as a linear combination of complex exponentials. For arbitrary compact groups there is also a natural notion of band-limited function. In this case computation of a Fourier transform requires a quadrature law or sampling theorem as well as an efficient algorithm for computing the Fourier coefficients from the samples. The latter sometimes involves the use of fast orthogonal polynomial transforms, which are discussed separately in Section 5. We close with some open questions in Section 6.

Because this paper is a survey, it is largely expository in nature. Pointers to more thorough discussions of the material are given throughout the paper. Our particular focus has caused us to omit other important generalizations of the FFT. Perhaps the most noticeable omission is a discussion of the wealth of exciting advances in the area of wavelets, for which either of the works [18, 27] would be a great place for the interested reader to begin.

**Acknowledgements.** We would like to thank Larry Finkelstein and Bill Kantor for inviting us to participate in the DIMACS workshop on groups and computation. We would also like to thank Michael Clausen for several helpful suggestions.

## 2. Two origins for the FFT

The generalizations discussed in this paper have as their natural predecessors some of the well-known abelian FFT algorithms. To date, abelian groups have provided the most useful class of FFTs, justifying the wealth of algorithms devoted to this special case. It would be well beyond the scope of this paper to discuss all of these approaches and we refer the interested reader to the books [38, 90] and their many references for this material.

Instead, to focus our attention, in this section we discuss just two of these earlier algorithms, both of which illustrate many features of a more general theory. Our first example is an algorithm due to the statistician Yates [93]. This was developed for the efficient analysis of data from $2^k$-factorial designs—a particular aspect of the statistical analysis for experimental designs. Our second example is the Cooley-Tukey FFT [26], probably the most famous of these abelian techniques. Here too the motivation came from the need for efficient data analysis, in this case data from time series. Both algorithms may be formulated as solutions to computational problems which have naturally suggested group theoretic formulations.

2.1. **Yates - Fast interaction analysis for $2^k$-factorial designs.** One version of the abelian FFT is due to the statistician and design theorist Yates. In order to efficiently compute the interaction analysis for data from a $2^k$-factorial design, Yates described an algorithm which, as we will see, is an FFT for the group $(\mathbf{Z}/2\mathbf{Z})^k$.

As a set, a **$2^k$-factorial design** is simply the set of all $k$-tuples of signs $\{+1,-1\}^k$, which can be thought of as the vertices of the $k$-dimensional hypercube or the space of binary $k$-tuples. In the context of statistical design it is a natural way of indexing the trials of an experiment which depends on $k$ **factors**, each of which may be set at a **high** or **low** level. For example, consider the following toy data set for a $2^3$-factorial design. Here we could imagine the following scenario: A farmer is interested in factors affecting the growth of his wheat. In particular he'd like to understand how sunlight, weed killer and fertilizer affect the height of the plant. Simplifying things, the plants are exposed to all three factors at various combinations of two possible levels, high and low, denoted as $+$ or $-$ respectively. Each of the eight possible combinations is applied to the same number of plants and finally the average height, denoted $\alpha_{swf}$ for a given choice of sunlight ($s$), weed killer ($w$) and fertilizer ($f$), at each combination is recorded. Table 1 gives a possible summary of such an experiment. Thus, for example, row two in Table 1 indicates that

| $s$ | $w$ | $f$ | $\alpha_{swf}$ |
|---|---|---|---|
| $+$ | $+$ | $+$ | 69 |
| $-$ | $+$ | $+$ | 81 |
| $+$ | $-$ | $+$ | 63 |
| $-$ | $-$ | $+$ | 77 |
| $+$ | $+$ | $-$ | 61 |
| $-$ | $+$ | $-$ | 92 |
| $+$ | $-$ | $-$ | 54 |
| $-$ | $-$ | $-$ | 89 |

TABLE 1. An example of a possible data set for a $2^3$-factorial design.

at a low level of sunlight, and high levels of weed killer and fertilizer, the average wheat plant height was 81 centimeters.

Our farmer is interested in the various effects of the factors, both individually and in combinations. There are various quantities which seem to be worth examining. The zeroth order effect is the **grand mean** or total average height, denoted $\mu_{gr}$. This is simply the average of all the (already averaged) heights

$$\mu_{gr} = \frac{1}{8} \sum_{(s,w,f)\in\{+,-\}^3} \alpha_{swf}.$$

The grand mean estimates the intrinsic yield of simply growing wheat, *i.e.,* by virtue of planting wheat, the average height which could be expected independent of doing anything to the plant. Next to be considered might be the "pure" first order effects: the effect of one particular factor, all other factors being held equal. In the case of sunlight this might be measured by considering the differences of the average yields at a high level of sunlight versus the average at a low level

$$\mu_S = \frac{1}{4}(\alpha_{+--} + \alpha_{+-+} + \alpha_{++-} + \alpha_{+++}) - \frac{1}{4}(\alpha_{---} + \alpha_{--+} + \alpha_{-+-} + \alpha_{-++}).$$

Pure second order and third order effects have similar descriptions, the collection of which may be coded up as the computation of the following matrix-vector multiplication

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} \alpha_{+++} \\ \alpha_{-++} \\ \alpha_{+-+} \\ \alpha_{--+} \\ \alpha_{++-} \\ \alpha_{-+-} \\ \alpha_{+--} \\ \alpha_{---} \end{pmatrix} = \begin{pmatrix} 8\mu_{gr} \\ 4\mu_S \\ 4\mu_W \\ 4\mu_{SW} \\ 4\mu_F \\ 4\mu_{SF} \\ 4\mu_{WF} \\ 4\mu_{SWF} \end{pmatrix} \quad (2.1)$$

When written in this fashion, it becomes evident that this analysis is the same as computing the projection of the data vector onto an orthogonal basis in which the projections have a natural interpretation.

At this point group theory may enter. The data vector is considered as an element in the vector space of complex-valued functions on the group $(\mathbf{Z}/2\mathbf{Z})^3$. Initially the data is expressed in terms of the basis of delta functions on the group,

$$\alpha = \sum_{x \in (\mathbf{Z}/2\mathbf{Z})^3} \alpha(x)\delta_x .$$

Posed in this way, analysis of the data is a rewriting of the data vector in terms of a new basis for which the coordinates seem to carry more information. In this case, the new basis is precisely the basis of characters (one-dimensional representations) of $(\mathbf{Z}/2\mathbf{Z})^3$, and the computation of the matrix vector product above is the Fourier transform of the data.

Notice that if computed directly, $8^2$ operations are required to compute the full analysis in the above example. Analogous decompositions may be obtained for any $2^k$-factorial design and in general $(2^k)^2$ operations are required to compute the analysis directly. For $k$ large this cost is prohibitive.

Yates succeeded in finding an algorithm which is much more efficient than direct computation. More precisely, he discovered an algorithm which requires at most $3 \cdot 2^k \cdot k = 3 \cdot 2^k \log(2^k)$ operations [93]. To give a brief, group theoretic interpretation of his algorithm, let $H_k$, denote the matrix of the Fourier transform on $(\mathbf{Z}/2\mathbf{Z})^k$. Thus $H_1$ is the $2 \times 2$ matrix

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and $H_3$ is the matrix appearing in equation (2.1). Any character of $(\mathbf{Z}/2\mathbf{Z})^3$ may be written as a tensor product of characters of the group $\mathbf{Z}/2\mathbf{Z}$. By factoring characters in this way it easily follows that $H_3 = H_1 \otimes H_1 \otimes H_1$, and hence that $H_3$ has the factorization

$$H_3 = [I_4 \otimes H_1] \cdot [I_2 \otimes H_1 \otimes I_2] \cdot [H_1 \otimes I_4]$$

where $I_j$ denotes the $j \times j$ identity matrix and $\otimes$ the usual tensor product of matrices. This is a sparse decomposition of the matrix $H_3$, and the Fourier transform of $\alpha$ is computed by multiplying by each of these sparse matrices in turn.

We shall see other algorithms based on factoring representations as tensor products later in the paper. This method is useful for computing Fourier transforms on a direct products of groups, and on solvable groups.

*Remark 2.1.* The Fourier transform we have just described on the group $(\mathbf{Z}/2\mathbf{Z})^k$ is also known as the Walsh-Hadamard transform.

2.2. **Cooley-Tukey - A fast algorithm for time series analysis.** Cooley and Tukey's (re-)discovery of the FFT was motivated by the need to efficiently analyze data from time series. Loosely speaking, this is data—sometimes called the "signal"—which is (continuously) indexed by time. Standard analysis of such data rewrites the function as a Fourier series, *i.e.,* as a linear combination of sines and cosines, families of periodic functions which have a well-understood behavior. The particular application that Cooley and Tukey had in mind early in the 1960's was the analysis of seismic data, potentially monitoring seismic activity within the Soviet Union. Efficient algorithms were needed to detect nuclear tests, thereby avoiding the need for site visits, which at that time were a sticking point in the negotiation of a nuclear test ban treaty. In general, long time series would be obtained and fast algorithms were mandatory for a useful analysis (cf., [24, 25]). We proceed by presenting a fairly standard treatment of the algorithm and then giving it a group theoretic reinterpretation.

2.2.1. *The Cooley-Tukey algorithm.* The discrete Fourier transform (DFT) of a sequence of $N$ complex numbers, $x_0, x_1, \ldots, x_{N-1}$ is the sequence

$$X_k = \sum_{j=0}^{N-1} x_j \omega^{j\,k}, \qquad k = 0, 1, 2, \ldots, N-1; \qquad \text{for} \quad \omega = e^{2\pi i/N}. \tag{2.2}$$

If computed directly, each number $X_k$ would require $N$ operations which gives a total of $N^2$ operations to compute the DFT. As $N$ becomes large this cost quickly becomes prohibitive. Cooley and Tukey derived and implemented an algorithm which given a prime factorization of $N = p_1 \cdots p_r$, computed the DFT in $N \sum_i p_i$ operations [26]. If each $p_i = 2$, this is $2N \log_2 N$ operations. This algorithm, together with many variants as well as different approaches to computing the DFT, yield a family of techniques which for any $N$ give an $O(N \log N)$ algorithm for computing the DFT (see, e.g., [19, 38, 90]). This speed-up has had tremendous significance, effectively making digital signal processing a reality.

To see how this algorithm works, let us consider what happens when $N = pq$ has factors. In this case we change the indexing of the sequences $x_j$ and $X_k$ by setting

$$j = i_2 + i_1 q, \qquad k = m_1 + m_2 p \tag{2.3}$$

and then defining the two-dimensional arrays

$$x_{i_1,i_2} = x_j\,, \qquad i_1 = 0, \ldots p-1, \qquad i_2 = 0, \ldots q-1$$

and

$$X_{m_1,m_2} = X_k, \qquad m_1 = 0, \ldots p-1, \qquad m_2 = 0, \ldots q-1$$

Substituting into (2.2) we obtain

$$X_{m_1,m_2} = \sum_{i_2=0}^{q-1} \omega^{i_2(m_1+m_2 p)} \sum_{i_1=0}^{p-1} (\omega^q)^{i_1 m_1} x_{i_1,i_2}. \tag{2.4}$$

The computation may then be performed in two steps. First, $q$ transforms of length $p$ are computed according to

$$\tilde{X}_{i_2,m_1} = \sum_{i_1=0}^{p-1} (\omega^q)^{i_1 m_1} x_{i_1,i_2}.$$

Next, $p$ transforms of length $q$ are computed,

$$\sum_{i_2=0}^{q-1} \omega^{i_2(m_1+m_2p)} \tilde{X}_{i_2,m_1}.$$

Notice that instead of $(pq)^2$ operations, the above uses $(pq)(p+q)$ operations. If we had more factors in the expression (2.4), then this approach would work even better, giving Cooley and Tukey's result. The main idea is that we have converted a one-dimensional algorithm, in terms of indexing, into a two-dimensional algorithm.

2.2.2. *Group theoretic interpretation of the FFT.* Let $G = \mathbf{Z}/N\mathbf{Z}$ be the group of integers modulo $N$. Then the characters of $G$ are the functions $\zeta_0, \ldots, \zeta_{N-1}$ defined by $\zeta_k(j) = \omega^{jk}$. If we view the sequence $x_j$ as defining a function on $\mathbf{Z}/N\mathbf{Z}$ and the sequence $X_k$ as defining a function on the group of characters, then the discrete Fourier transform is seen to be the Fourier transform on $\mathbf{Z}/N\mathbf{Z}$. In other words, it is given by the sums

$$X_k = X(\zeta_k) = \sum_{j \in \mathbf{Z}/N\mathbf{Z}} \zeta_k(j) x_j.$$

If $N = pq$, then the indexing used above in the Cooley-Tukey algorithm can be explained in terms of the subgroup, $q\mathbf{Z}/N\mathbf{Z}$, generated by the group element $q$. Consider the reindexing we used for the group element $j$ via the expression

$$j = i_2 + (i_1 q).$$

As $j$ is now viewed as an element of the group $\mathbf{Z}/N\mathbf{Z}$, this equation represents a "factorization" of $j$ as the sum of the group elements $i_2$ and $i_1 q$. The elements $i_1 q$ are precisely the elements of the subgroup $q\mathbf{Z}/N\mathbf{Z}$, and the elements $i_1$ form a complete set of coset representatives for $\mathbf{Z}/N\mathbf{Z}$ relative to this subgroup.

The index $m_1$ has a different interpretation, in terms of restrictions of characters to the subgroup $q\mathbf{Z}/N\mathbf{Z}$. This subgroup has characters $\chi_m(i_1 q) = (\omega^q)^{i_1 m}$, and the restriction of $\zeta_k$ to $q\mathbf{Z}/N\mathbf{Z}$ is the character $\zeta_k \downarrow q\mathbf{Z}/N\mathbf{Z} = \chi_{m_1}$. The quantity $\tilde{X}(i_2; m_1)$ is therefore indexed by pairs consisting of a coset representative for $(\mathbf{Z}/N\mathbf{Z})/(q\mathbf{Z}/N\mathbf{Z})$ and a character of $q\mathbf{Z}/N\mathbf{Z}$. If we let $A$ denote the set of coset representatives, then the expression (2.4) for the Fourier transform used to derive the FFT can be rewritten as follows

$$X(\zeta) = \sum_{a \in A} \zeta(a) \sum_{b \in q\mathbf{Z}/N\mathbf{Z}} (\zeta \downarrow q\mathbf{Z}/N\mathbf{Z})(b) x_{a+b}$$

where $\zeta$ is any character of $\mathbf{Z}/N\mathbf{Z}$. In this form, both (2.4) and the Cooley-Tukey algorithm generalizes to any finite group with a proper nontrivial subgroup (see Section 3). If we want to derive the Cooley-Tukey algorithm for larger factorization $N = p_1 \cdots p_r$, we simply apply this technique to a chain of subgroups of $\mathbf{Z}/N\mathbf{Z}$.

The indexing scheme for characters in $\mathbf{Z}/N\mathbf{Z}$ may be nicely described by a diagram indicating the restrictions of the irreducible representations from group to subgroup within the chain $\mathbf{Z}/N\mathbf{Z} > q\mathbf{Z}/N\mathbf{Z} \cong \mathbf{Z}/p\mathbf{Z} > 1$. As an example, the diagram for $\mathbf{Z}/6\mathbf{Z} > 2\mathbf{Z}/6\mathbf{Z} \cong \mathbf{Z}/3\mathbf{Z} > 1$ is shown in Figure 1. In this diagram there is a unique path of length two from 1, at the right hand side, to $\zeta_k$ at the left. Thus the representations of $\mathbf{Z}/6\mathbf{Z}$ may be indexed by paths of length 2. The characters of $2\mathbf{Z}/6\mathbf{Z}$ correspond to paths of length one starting at the left. If we are given a character $\zeta_k$ of $\mathbf{Z}/6\mathbf{Z}$ determined by a path of length two, then the path corresponding to the restriction may be obtained by forgetting the last edge of the

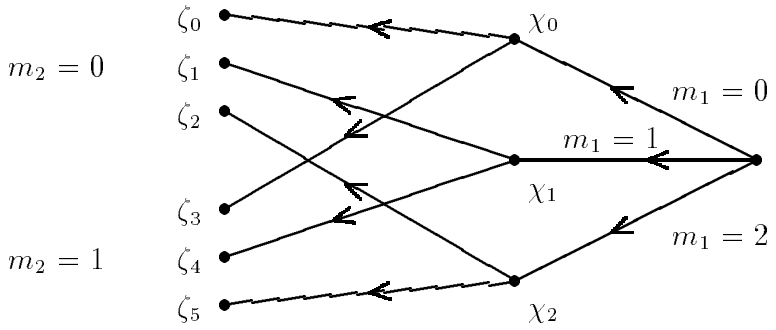



FIGURE 1. The Bratteli diagram for $\mathbf{Z}/6\mathbf{Z} > 2\mathbf{Z}/6\mathbf{Z} > 1$.

path. This determines $m_1$. Although $m_2$ can be described using this diagram, this index is not needed by the algorithm, as it only appears in conjunction with $m_1$. When we come to treat the Fourier transforms of functions on nonabelian finite groups, we will see that the values of the Fourier transform of a function are no longer indexed by representations, but they may still be indexed by paths in a diagram reflecting the restrictions of irreducible representations of subgroups occurring in a chain of subgroups.

*Remark 2.2.* The paper [3] gives a different representation theoretic interpretation of Cooley-Tukey, relating it to the Weil-Brezin map.

2.2.3. *The transition from continuous to discrete.* Many data sets, e.g., seismic data, come from a continuous signal from which we have taken a finite set of samples. This means that we must conduct our analysis with incomplete data; the signal is a function on the real line, but we can only compute Fourier transforms of a finite sequence. To relate the Fourier transform of a continuous function to the transform of a finite sequence, we should make the following assumptions:

- The signal is a periodic function on the real line.
- The signal is **band-limited**. *I.e.* , the signal has a finite Fourier expansion.

Notice that the first assumption implies that we may view the signal as a function on the circle, and thus the second assumption makes sense. In practice, neither of these assumptions will be satisfied, but as we only ever encounter data over a finite time period, we might as well assume the signal has been extended to a periodic function of the line. The second assumption is on slightly firmer ground, as it is possible to bound the effect of higher frequencies on the sampling procedure, provided there is only a small part of the signal is at these frequencies.

Without loss of further generality, we may assume that the signal $f$ is periodic function on the real line of period one, so that the second assumption implies that $f$ has the expansion

$$f(t) = \sum_{k=-N}^{N} \hat{f}(k) e^{-2\pi i k t}$$

for $0 \leq t < 1$. The number $N$ is called the **bandwidth** of $f$. The Fourier coefficients $\hat{f}(k)$ are given by the formula

$$\hat{f}(k) = \int_0^1 f(t) e^{2\pi i k t} dt. \tag{2.5}$$

In order to compute the Fourier coefficients we need some way to reduce the integrals (2.5) to finite sums involving a discretization of $f$. Such results go by the name of quadrature or sampling theorems. If $f$ is band-limited with bandwidth $N$, then the following quadrature rule is well-known:

$$\hat{f}(k) = \frac{1}{2N+1} \sum_{j=0}^{2N} f\left(\frac{j}{2N+1}\right) e^{2\pi i k j/(2N+1)}. \tag{2.6}$$

Thus, the signal may be recovered from its finite collection of equispaced samples $\left\{f\left(\frac{j}{2N+1}\right) | 0 \leq j \leq 2N+1\right\}$, in the sense that this information is sufficient to compute its Fourier coefficients. If we define finite sequences, $x_j = \frac{1}{2N+1} f\left(\frac{j}{2N+1}\right)$, and $X_k = \hat{f}(k)$, then the quadrature rule (2.6) becomes a DFT of length $2N+1$, and may be computed efficiently by the Cooley-Tukey algorithm.

When used as above, the Cooley-Tukey FFT is an algorithm for computing the Fourier expansion of a band-limited function on the circle. For this there are two ingredients:

1. A sampling theorem which reduces the necessary integrals to finite sums.
2. A fast Fourier transform, which efficiently computes the discretized integrals.

We shall see later that both of these ingredients have natural generalizations to compact groups.

2.2.4. *Summary.* We have seen above that various tasks in data analysis required algorithms for the efficient projection of data vectors onto vectors which admit some natural interpretation for the problem at hand. In both cases these projections have a group theoretic interpretation as the projection of functions defined on a group, onto vectors derived from irreducible representations of the group. With this group theoretic interpretation, a natural direction for generalization arises: Given a function on a group with a finite expansion in terms of irreducible matrix coefficients, describe an algorithm to efficiently compute this expansion. The remainder of this paper is a description of some recent progress towards solving this problem.

## 3. FFTs for finite groups

The first generalization we pursue casts both Yates's algorithm and the Cooley-Tukey FFT as a particular instance of a fast Fourier transform on a finite group. This more general framework seems to have been first considered by Willsky [92] for uses in signal processing. Later motivations came from filter design [57] and data analysis [31].

**Definition 3.1** (Fourier Transform)**.** Let $G$ be a finite group and $f$ be a complex-valued function on $G$.

1. Let $\rho$ be a matrix representation of $G$ of dimension $d_\rho$, *i.e.,* $\rho$ is a group homomorphism from $G$ into the group of $d_\rho \times d_\rho$ invertible complex matrices.

Then the **Fourier transform** of $f$ at $\rho$, denoted $\hat{f}(\rho)$ is the matrix sum,

$$\hat{f}(\rho) = \sum_{s \in G} f(s)\rho(s). \tag{3.1}$$

2. Let $\mathcal{R}$ be a set of matrix representations of $G$. Then the **Fourier transform of $f$ on $\mathcal{R}$** is the collection of Fourier transforms of $f$ at the representations in $\mathcal{R}$.

**Fast Fourier transforms** or **FFTs** are algorithms for computing Fourier transforms efficiently.

**Example 3.2.** It is not difficult to see that both examples of Section 2 are special cases of this definition. In each case the irreducible representations are one-dimensional so there is no ambiguity about the choice of basis for the matrix representations. For $(\mathbf{Z}/2\mathbf{Z})^k$, the functions $\chi_v(w) = (-1)^{\langle v,w \rangle}$ for $v, w \in (\mathbf{Z}/2\mathbf{Z})^k$ (and inner product computed mod 2) give all irreducible representations. In the second case the functions $\zeta_j(k) = e^{2\pi ijk/N}$ are the irreducible representations of the group $\mathbf{Z}/N\mathbf{Z}$.

*Remark 3.3.* The definition of Fourier transform given above is most convenient for our purposes, but a number of equivalent versions are possible.

1. Computing the Fourier transform of $f$ at $\rho$ is equivalent to computing the collection of scalar transforms at the matrix coefficients $\rho_{ij}$, for $1 \leq i, j \leq d_\rho$

$$\hat{f}(\rho)_{ij} = \sum_{s \in G} f(s)\rho_{ij}(s).$$

2. It is well-known that the matrix entries of a complete set of inequivalent irreducible matrix representations of $G$ form a basis for the vector space of complex functions on $G$. Computing the Fourier transform of a function $f$ at such a complete set of representations is equivalent to expanding the function $f^\vee(s) = f(s^{-1})$ in the basis of matrix coefficients. We shall simply refer to such a calculation as the computation of a Fourier transform.
3. The Fourier transform at a complete set of irreducible representations is equivalent to change of basis in the group algebra $\mathbf{C}[G]$, from the basis of point-masses (delta functions) to a basis of matrix coefficients.
4. The Fourier transform at the regular representation is a map from functions on the group to convolution operators on the group.
5. A **model representation** of $G$ is the direct sum of a complete set of irreducible representations of $G$. Computing a Fourier transform at a model representation is equivalent to computing a Fourier transform on the corresponding complete set of representations.

The arithmetic complexity of computing a Fourier transform conceivably depends on the choice of basis for the representations.

**Definition 3.4** (Group complexity)**.** Let $G$ be a finite group, and $\mathcal{R}$ any set of matrix representations of $G$. The **complexity of the Fourier transform** for the set $\mathcal{R}$, denoted $T_G(\mathcal{R})$, is defined to be the minimum number of operations needed to compute the Fourier transform of $f$ on $\mathcal{R}$ via a straight-line program for an arbitrary complex-valued function $f$ defined on $G$. Define the **complexity of the group** $G$ to be

$$\mathcal{C}(G) = \min_{\mathcal{R}} \{T_G(\mathcal{R})\}$$

where $\mathcal{R}$ varies over all complete sets of inequivalent irreducible matrix representations of $G$.

*Remark 3.5.*
1. The computational model used here is a common one in which an operation is defined as a single complex multiplication followed by a complex addition.
2. Note that $T_G(\mathcal{R}) = T_G(\oplus_{\rho \in \mathcal{R}} \rho)$. *I.e.*, the complexity for a set of representations is equal to the complexity for the direct sum of the representations.

The complexity of a finite group provides a classification of finite groups according to the complexity of the most efficient algorithm to compute some such transform on the group. Direct computation of any Fourier transform gives the upper and lower bounds

$$|G| - 1 \leq \mathcal{C}(G) \leq T_G(\mathcal{R}) \leq |G|^2$$

In [68] we have introduced the related quantity $t_G(\mathcal{R})$, called the **reduced complexity** and defined by

$$t_G(\mathcal{R}) = T_G(\mathcal{R}) / |G| \tag{3.2}$$

This definition simplifies the statements and proofs of many following results.

**The separation of variables approach.** The separation of variables approach is a method of constructing fast Fourier transform algorithms on arbitrary finite groups, which generalizes the Cooley-Tukey method for cyclic groups. Aspects of this approach to computing Fourier transforms have previously appeared [12, 22, 31], but it has only recently been given a general formulation [68, 69, 70]. (See also [23] for results in this direction.)

The basic idea is to re-index the calculation so as to replace the single sum (3.1) defining a Fourier transform by a multiple sum over many different "coordinates". To do this, we find group and representation theoretic interpretations for the substitution (2.3), thereby generalizing the ideas of Section 2.2 to other classes of finite groups. Once our sum is in this multi-dimensional form, we apply ideas analogous to those in multiple integration; factoring terms that don't involve a particular index through the individual sums, and computing the sums coordinate by coordinate. In this way, both new and previously discovered FFTs are obtained, but in a uniform fashion.

The separation of variables approach has a few main ingredients. The essential ones are the factorization of group elements and the use of adapted sets of representations, or Gel'fand-Tsetlin bases. These tools permit the rewriting of the transform as a multiple sum, which can be computed summing over one coordinate at a time. A careful choice of the factors of the group elements and the use of Schur's Lemma allows even further simplification of these expressions. These ideas can be applied either at the level of matrices (Section 3.1) or at the scalar level (Section 3.3), in which the Fourier transform is viewed as a collection of individual transforms at the matrix coefficients. Because scalar multiplication is commutative, whereas matrix multiplication is not, the scalar approach is much more flexible, and allows the summations to be performed in different orders. The added flexibility does not come for free, but is obtained at the expense of a slightly complicated indexing scheme for the matrix coefficients, which expresses the computation in terms of paths in Bratteli diagrams indexing the Gel'fand-Tsetlin basis.

Not all FFT algorithms are described by the separation of variables approach. In Section 3.4 we summarize alternative approaches for abelian groups which proceed by relating the Fourier transforms of groups which are not in a group-subgroup relationship. In Section 3.5 we consider the effect of having a normal subgroup. In that case the separation of variables technique is supplemented by factoring the group representations with tensor products. This is a key idea behind the fast transforms for abelian group extensions [83], solvable groups [12, 19], and supersolvable groups [7].

## 3.1. **Separation of variables at the matrix level.**

3.1.1. *The main idea.* Let $G$ be a finite group, $f$ a complex-valued function on $G$, and let $\rho$ be a matrix representation of $G$. To construct an FFT algorithm on $G$, proceed as follows. Assume that that each group element $g$ has a factorization of the form $g = a_n \cdots a_1$. Substitution of this into the definition of a Fourier transform (3.1) and the use of the homomorphism property of group representations gives

$$\begin{aligned}\hat{f}(\rho) &= \sum_{g=a_n \cdots a_1} \rho(a_n) \cdots \rho(a_1) f(a_n \cdots a_1) \\ &= \sum_{a_n} \rho(a_n) \cdots \sum_{a_2} \rho(a_2) \sum_{a_1} \rho(a_1) f(a_n \cdots a_1). \end{aligned} \tag{3.3}$$

The transform is now in a multi-dimensional form, and so can be computed by summing on $a_1$ first, then summing on $a_2$, and so on. An algorithm for computing Fourier transforms constructed in this way is a **matrix separation of variables algorithm**. Unfortunately, this trick alone does not give a fast Fourier transform. For this, the key point is to use the freedom of choice in the factorizations and the representations. We make these choices with the goal of obtaining matrices $\rho(a_i)$ with of a special form, e.g., block diagonal, or block scalar, or both.

To illustrate how this could lead to an efficient algorithm, suppose that the matrices $\rho(a_i)$ are all block diagonal with the blocks increasing in size with $i$. We start with the matrix-valued function $F_0(a_n, \ldots, a_1) = f(a_n, \ldots, a_1) \cdot I$, where $I$ denotes the identity matrix. This function depends on all coordinates, $a_1, \ldots, a_n$. After summing over the first $i$ coordinates, we obtain a function $F_i(a_n, \ldots a_{i+1})$, which has the recursive definition

$$F_i(a_n, \ldots a_{i+1}) = \sum_{a_i} \rho(a_i) F_{i-1}(a_n, \ldots, a_i). \tag{3.4}$$

As $i$ increases, the matrices occurring in equation (3.4) have blocks of increasing size, but the number of variables involved in the definition of $F_i$ decreases. By playing these two factors off against each other, efficient algorithms can be obtained.

In practice the matrices $\rho(a_i)$ are not only block diagonal, but may also have some blocks repeated, or also have a block scalar structure. In most cases, the redundancy coming from repeated blocks is the main effect giving an efficient algorithm. It is important here that the structure of the block matrices occurring are compatible. All this is achieved by using subgroup-adapted sets of representations, or equivalently, Gel'fand-Tsetlin bases.

*Remark 3.6.* The above is a brief synopsis of the papers [70, 68]. Special cases of the matrix separation of variables approach also occur in work of Clausen on FFTs for the symmetric group, cf. [23], especially Section 10. Orszag discusses an analogous coordinate by coordinate approach for computing eigenfunction transforms in [75].

Auslander, et. al. discuss how in the abelian case, choice of coset representatives can influence running times for various architectures [4].

3.1.2. *Adapted representations and structured matrices.*

**Definition 3.7** (Subgroup-adapted representations)**.** Let $G$ be a finite group, let $\mathcal{R}$ be a set of matrix representations of $G$, and let $H$ be a subgroup of $G$. If $\rho$ is a representation of $G$, let $\rho \downarrow H$ denote the representation of $H$ obtained by restricting $\rho$ to $H$. We say that $\mathcal{R}$ is $H$**-adapted** if there is a set $\mathcal{R}_H$ of inequivalent irreducible matrix representations of $H$ such that the set of restricted representations

$$(\mathcal{R} \downarrow H) = \{\rho \downarrow H \mid \rho \in \mathcal{R}\}$$

is a matrix direct sum of representations in $\mathcal{R}_H$.

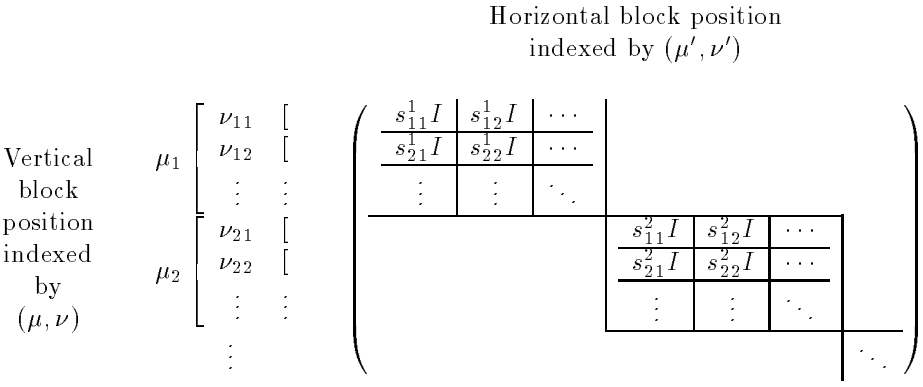
Thus a set of representations is adapted to a subgroup $H$ if the restrictions to $H$ of the representations in the set have block diagonal form according to their decompositions into irreducible representations of $H$. A set of representations is said to be **adapted to a chain of subgroups** if it is adapted to each subgroup in the chain.

Gel'fand-Tsetlin bases are an equivalent, and sometimes more useful, way of formulating this concept. A basis for a representation space is called a **Gel'fand-Tsetlin basis** relative to the subgroup $H$ if the matrix representation obtained by expressing the representation in coordinates for this basis is adapted. Systems of Gel'fand-Tsetlin bases for collections of representations are defined similarly.

*Remark 3.8.* Sometimes it it more convenient to deal with just a single representation, rather than a whole set $\mathcal{R}$ of representations. In this case the simple trick of taking the direct sum $\Delta = \oplus_{\rho \in \mathcal{R}} \rho$, allows us to relate many results about single representations to results about sets of representations. In particular, note that $\mathcal{R}$ is $H$-adapted if and only if $\{\Delta\}$ is $H$-adapted.

In terms of these definitions, we can reformulate Schur's Lemma in the following fashion.

**Lemma 3.9** (Schur's Lemma)**.** *Assume $G \geq K \geq H$ is a chain of finite groups, $a \in K$ centralizes $H$, and $\rho$ is an irreducible representation of $G$, which is adapted to both $K$ and $H$. Then $\rho(a)$ has the following block diagonal/block scalar form.*




$\rho \downarrow K = \mu_1 \oplus \mu_2 \oplus \ldots$, *where the $\mu_i$ are irreducible representations of $K$.*

$\mu_i \downarrow H = \nu_{i1} \oplus \nu_{i2} \oplus \ldots$, *where the $\nu_i$ are irreducible representations of $H$.*

*The indices, $\mu, \mu'$ and $\nu, \nu'$ range over the sets, $\{\mu_i\}$ and $\{\nu_{ij}\}$, consisting of those representations occurring in the restriction of $\rho$ to $K$ and $H$ respectively. The $s^i_{jk}$ are scalar. There is a nonzero scalar block at position $(\mu, \nu)$, $(\mu', \nu')$ only if $\mu = \mu'$ and $\nu = \nu'$.*

In this form, Schur's Lemma says that the representation matrices of elements of $K$ centralizing $H$ are sparse. Multiplying two such matrices can be done very efficiently, so if the elements we choose in the factorizations for the separation of variables algorithm have this form, then we should be able to construct a fast Fourier transform. To make this more precise, we need a measure of how sparse these matrices are. For this, we set up the following notation.

Assume $K \geq H$ are finite groups with complete sets of inequivalent irreducible representations $\hat{K}$ and $\hat{H}$ respectively. Then let

$$\begin{aligned}\mathcal{M}(K,H) &= \text{The maximum multiplicity occurring in the restriction of} \\ &\qquad \text{representations from } K \text{ to } H \\ &= \max\{\langle \mu \downarrow H, \nu\rangle : \mu \in \hat{K}, \nu \in \hat{H}\} \\ &= \text{The maximum possible number of nonzero entries in a row of } \rho(a).\end{aligned}$$

**Corollary 3.10.** *Assume $G \geq K \geq H$ and let $a \in K$ centralize $H$.*

1. *Let $\rho$ be a representation of $G$ adapted to $G, K$, and $H$, and let $F$ be any $d_\rho \times d_\rho$ matrix. Then $\rho(a) \cdot F$ can be computed in $\mathcal{M}(K,H) \cdot d_\rho^2$ scalar operations.*
2. *Let $\Delta$ be a model representation,* i.e., *the direct sum of a complete set of inequivalent irreducible representations of $G$, and let $F = \sum_{g \in G} \Delta(g) f(g)$ for any complex-valued function $f$ on $G$. Assume $\Delta$ is adapted to $G, K$, and $H$. Then $\Delta(a) \cdot F$ may be computed in $\mathcal{M}(K,H)\,|G|$ scalar operations.*

3.1.3. *A typical example: The symmetric group.* Given Corollary 3.10, the application of the separation of variables idea only involves some simple counting - assuming the multiplicities are easy to compute. To illustrate, we now rederive an algorithm due to Clausen [22] for computing Fourier transforms on the symmetric group.

To construct an FFT for the symmetric group, we will need to use representations adapted to the chain of subgroups

$$S_n > S_{n-1} > \cdots > S_1 = \{1\} \tag{3.5}$$

where $S_j < S_n$ is the subgroup fixing pointwise the elements $j+1, \ldots, n$. Such a basis is uniquely determined up to scale factors. Two common choices are Young's orthogonal and Young's seminormal forms (see eg. [55]).

**Theorem 3.11** (Clausen [22], Theorem 1.4)**.** *The Fourier transform on $S_n$ may be computed at a complete set of irreducible representations adapted to the chain of subgroups* (3.5) *in no more than $\frac{(n+1)n(n-1)}{3} \cdot n!$ scalar operations.*

*Proof.* According to our general philosophy, we factor the elements of the symmetric group using the pairwise adjacent transpositions $t_2, \ldots, t_n$, where $t_j$ denotes the transposition $(j-1, j)$. Specifically, if we let $Y_n$ denote the set of words of length $n-1$,

$$\{t_2 \cdot t_3 \cdots t_n, e \cdot t_3 \cdots t_n, \ldots, e \cdot e \cdots e \cdot t_n, e \cdot e \cdots e\}$$

where $e$ denotes the identity, then the factorizations needed for the separation of variables algorithm are those in the sets, $Y_n, Y_{n-1}, \ldots, Y_2$ of words of length $\binom{n}{2}$.

Now use the separation of variables algorithm for this set of words. *I.e.*, calculate the Fourier transform of a model representation $\Delta$ by first summing over the right-most factor, then the next right-most and so on. This process naturally breaks up into $n-1$ sections as we sum over those factors coming from each of the sets $Y_k$.

Consider now the summations and multiplications for the factors coming from $Y_k$. In these summations, all the matrices occurring have the form of Fourier transforms of functions on $S_k$, and are therefore block diagonal according to the restriction of $\Delta$ to $S_k$. Computing one block at a time, the multiplications that occur satisfy the conditions of Corollary 3.10 with $G = S_k$, $a = t_i$ for some $i$, $K = S_i$ and $H = S_{i-2}$. Young's Rule (see e.g. [55]) shows that $\mathcal{M}(S_i, S_{i-2}) = 2$. So the matrix multiplication by $\Delta(t_i)$ on the left that occurs at this stage takes only $2\,|S_k|$ scalar operations. The matrix additions that occur do not add to the overall operation count. If we now total the operations occurring in the separation of variables algorithm, and take into account that multiplying by the identity can be done for free, then we obtain the following count

$$\sum_{k=2}^{n} \frac{|S_n|}{|S_k|} \sum_{i=0}^{k-2} (2\,|S_k|) \cdot (k-i-1) = \frac{(n+1)n(n-1)}{3}\,|S_n|\,. \tag{3.6}$$

□

3.1.4. *General results and further examples.* The example of the symmetric group illustrates most of the general features of the matrix separation of variables technique. In particular, the sets $Y_k$ appearing in the proof are factorizations of sets of coset representatives for the successive subgroup pairs occurring in the original subgroup chain. It is common, in this approach, to split the algorithm in this way, and the cancellations that occur in the sum (3.6) are then valid in general. This sort of estimate and analysis motivates the use of the $t_G$ notation. The approach of using coset representatives together with a separation of variables type algorithm appears in [31, 23, 19].

To state a fairly general result of this type, we introduce an extension of the multiplicity notation. If $G$ is a finite group, $K_n \geq \cdots \geq K_0$ is a chain of subgroups, and $g$ is any element of $G$, then let $\mathcal{M}(g) = \mathcal{M}(K, H)$ for $K$ the smallest subgroup in the chain containing $g$, and $H$ the largest subgroup in the chain centralizing $g$.

**Theorem 3.12** ([68], Corollary 4.9)**.** *Let $K_n \geq \cdots \geq K_0$ be a chain of finite groups, let $\mathcal{R}$ be a complete set of inequivalent irreducible representations of $K_n$ adapted to the chain, and let $X$ be a set of words representing the factorizations of a complete set of coset representatives of $K_n/K_{n-1}$. Let $\gamma$ be the maximum length of any word in $X$. Then*

$$t_{K_n}(\mathcal{R}) \leq t_{K_{n-1}}(\mathcal{R}_{K_{n-1}}) + \sum_{k=0}^{\gamma-1} \sum_{s_n \cdots s_0 \in \tilde{X}_k} \mathcal{M}(s_0) \tag{3.7}$$

*where $\tilde{X}_k$ is obtained from $X$ by deleting $k$ elements from the right of each word and then deleting all words with an identity element at the far right.*

This result is by no means the best of its type. On the one hand, we could be more careful about counting the exact number of operations appearing in the matrix multiplications. This is done in [68], where we do a careful analysis of all the possible matrix multiplications that can occur, and how the structures of the

block matrices interact with each other. Generalizing in another direction, we can place some conditions on the types of factorizations that we allow, and thus obtain succinct results for an arbitrary finite group (cf. Section 3.2).

Theorem 3.12 gives us FFTs for a number of interesting groups: the classical Weyl groups, wreath products of $S_n$, and classical finite groups of Lie type. As an example, we state a result for the general linear group although in the next section we will improve on it.

Let $GL_n(q)$ denote the group of invertible $n \times n$ matrices with entries in the field of $q$ elements where $q$ is a prime power. Let $P_n$ be the parabolic subgroup of $GL_n$ of all block matrices of the form

$$\left(\begin{array}{c|c} A & B \\ \hline 0 \ \ldots \ 0 & C \end{array}\right) \tag{3.8}$$

where $A \in GL_{n-1}(q)$, $B \in \mathbb{F}_q^{n-1}$ and $C \in \mathbb{F}_q^{\times}$.

**Theorem 3.13.** *Let $n \geq 2$, let $q$ be a prime power, and let $\mathcal{R}$ be a complete set of inequivalent irreducible representations of $GL_n(q)$ adapted to the chain of subgroups*

$$GL_n > P_n > GL_{n-1} \times GL_1 > GL_{n-1} > \cdots > GL_1.$$

*Then we have*

$$\mathcal{C}(GL_n(q)) \leq T_{GL_n(q)}(\mathcal{R}) < \frac{1}{2} 2^{2n} q^{2n-2} \left|GL_n(q)\right|.$$

The proof of this result follows the same lines as for the symmetric group, using a result of Thoma [88] to bound the multiplicities. Theorem 3.12 merely serves to organize the operation count. A similar procedure goes through for the finite groups of Lie type, using the recent work of Hagedorn [45] for bounding the multiplicities.

*Remark 3.14.* Linton, Michler and Olsson [63] give an algorithm to compute the Fourier transform on $GL_n(q)$ using certain monomial model representations. Theorem 3.13 improves on this result.

3.1.5. *Homogeneous spaces.* A nice feature of this approach is that we obtain results for homogeneous spaces with no extra effort. If $G$ is a finite group, and $H$ is a subgroup, then specifying a function on the homogeneous space $G/H$ is equivalent to specifying a right $H$-invariant function on $G$. Using this correspondence, we can define the Fourier transform of a function on $G/H$ and pose the same questions about the complexity of such a transform. Under appropriate conditions, an analogously defined reduced complexity, $t_{G/H}$, obeys the same kinds of recurrence relations as $t_G$. More specifically, with the right definitions, equation (3.7) holds with $t_{K_n}$ replaced by $t_{K_n/K_j}$, and $t_{K_{n-1}}$ replaced by $t_{K_{n-1}/K_j}$. Using this correspondence, results for groups transfer immediately to results for homogeneous spaces. As an example, we obtain the following result for the symmetric group. (See [68] for further examples.)

**Theorem 3.15.** *The Fourier transform of a complex function on $S_n/S_{n-k}$ may be computed at a complete set of irreducible representations of $S_n$, adapted to the chain of subgroups* (3.5) *in at most*

$$k\left(n^2 - kn + \frac{k^2-1}{3}\right)\left|\frac{S_n}{S_{n-k}}\right|$$

*scalar operations.*

This result is only useful for $k \geq 3$, but other techniques may be used for $k = 1$.

3.2. **Finding the right factorizations.** The construction of a separation of variables type algorithm for computing Fourier transforms of functions on a finite group involves many choices. Not only do we choose a chain of subgroups, but we are also free to choose the factorizations of each element of the group.

After choosing a chain of subgroups, Theorem 3.12 tells us how to construct an algorithm given only factorizations of coset representatives. In addition, it suggests that we choose factors for which the value of $\mathcal{M}$ is small, or equivalently, factors which commute with as large a subgroup of the chain as possible. However, this choice may conflict with the desire to use a long chain of subgroups, but still have short factorizations of group elements.

The general problem of finding an optimal set of factorizations seems difficult to solve, but still we can find partial solutions. An important step in simplifying the discussion is to separate the different contributions to the complexity of our algorithm coming from the length of subgroup chains, the length of factorizations, and the multiplicities $\mathcal{M}(s)$. We do this by considering a weaker upper bound on the complexity which is a trivial corollary of Theorem 3.12. Although this does not solve the problem of finding optimal FFTs, it does in practice give useful guidelines, and perhaps most importantly, it provides a place to start in the search for the best factorizations.

3.2.1. *Strong generating sets and adapted diameters.*

**Definition 3.16.** Assume $G$ is a finite group, $S$ is a subset of $G$, and

$$\mathcal{K} = \{K_n \geq K_{n-1} \geq \cdots \geq K_0\} \tag{3.9}$$

is a chain of subgroups of $G$.

1. The subset $S \subset G$ is said to be a **generating set for the chain of subgroups** if $S \cap K_j$ generates a set of coset representatives for $K_j/K_{j-1}$ for each $j \geq 1$.
2. The subset $S \subset G$ is a **strong generating set** for $G$, relative to the chain of subgroups, if $G = K_n$ and $K_0 = 1$ and $S$ is a generating set for the chain of subgroups.
3. The **adapted diameter** of the chain, relative to $S$ is defined to be

   $$\gamma(S, \mathcal{K}) = \gamma_1 + \cdots + \gamma_n$$

   where $\gamma_j$ is the maximum length of a product of elements in $S \cap K_j$ that is required to construct the coset representatives of $K_j/K_{j-1}$, starting from the coset of the identity, *i.e.,*

   $$\gamma_j = \min \left\{ l \geq 0 : \bigcup_{0 \leq i \leq l} (S \cap K_j)^i . K_{j-1} = K_j \right\}.$$

Note that $S$ is a strong generating set for a chain of subgroups if and only if its adapted diameter is finite.

The condition that a set of group elements $S$ is a strong generating set for the chain (3.9), is precisely the requirement that products of elements of $S$ may be used as the factorizations in a separation of variables algorithm based on the recursive application of Theorem 3.12. The adapted diameter of a chain $\mathcal{K}$ relative to $S$ is the smallest maximum length of factorization that could occur in such an algorithm.

The weaker complexity bound for the Fourier transform depends on the adapted diameter of a subset, on a factor describing how refined the subgroup chain is, and on a factor depending on the multiplicities of restriction of representations. Assume $G$ is a finite group, $S$ is a subset of $G$, and

$$\mathcal{K} = \{K_n \geq K_{n-1} \geq \cdots \geq K_0\}$$

is a chain of subgroups of $G$.

1. Let $\mathcal{M}(S, \mathcal{K}) = \max\{\mathcal{M}(s) : s \in S\}$, where $\mathcal{M}(s)$ is calculated relative to the chain $K_n \geq \cdots \geq K_0 \geq 1$.
2. Let $v(\mathcal{K}) = \max\{|K_j / K_{j-1}| : 1 \leq j \leq n\}$.

**Theorem 3.17.** ( [70] ) *Assume $\mathcal{K}$ is a chain of subgroups of length $n$, and $S$ is any subset of $G$. Let $\mathcal{R}$ be a complete set of irreducible representations of $K_n$, adapted to the chain. Then*

$$t_{K_n}(\mathcal{R}) \leq \gamma(S, \mathcal{K}) v(\mathcal{K}) \mathcal{M}(S, \mathcal{K}) + t_{K_0}(\mathcal{R}_{K_0}).$$

*If $K_0 = 1$ then $t_{K_n}(\mathcal{R}) \leq \gamma(S, \mathcal{K}) v(\mathcal{K}) \mathcal{M}(S, \mathcal{K})$.*

If $\mathcal{K}$ and $\mathcal{L}$ are two chains of subgroups of $G$, we say that $\mathcal{L}$ is a refinement of $\mathcal{K}$, if every subgroup in $\mathcal{K}$ is also in $\mathcal{L}$, and the largest and smallest subgroups of $\mathcal{L}$ are the same, respectively, as those of $\mathcal{K}$. The following easy lemma shows the effect of refining a subgroup chain.

**Lemma 3.18.** *Assume $G$ is a finite group, $\mathcal{K}, \mathcal{L}$ are chains of subgroups in $G$, and $S$ is a subset of $G$. If $\mathcal{L}$ is a refinement of $\mathcal{K}$, then $v(\mathcal{L}) \leq v(\mathcal{K})$ and $\mathcal{M}(S, \mathcal{L}) \leq \mathcal{M}(S, \mathcal{K})$, but $\gamma(S, \mathcal{L}) \geq \gamma(S, \mathcal{K})$.*

3.2.2. *Choosing the generating set.* Now we look at the problem of choosing the generating set $S$. We take the point of view that the factors $v(\mathcal{K})$ and $\mathcal{M}(S, \mathcal{K})$ in Theorem 3.17, are more important that $\gamma(S, \mathcal{K})$. This supposition is borne out in many examples, but it is also a convenient assumption. Whereas it is possible to simultaneously locally minimize $v(\mathcal{K})$ and $\mathcal{M}(S, \mathcal{K})$, it oftens seems impossible to simultaneously minimize either of these factors along with $\gamma(S, \mathcal{K})$. First we shall see how we may add elements to $S$ without decreasing $\mathcal{M}(S)$. Then, given a chain of subgroups $\mathcal{K}$ with $G = K_n$ and $K_0 = 1$, we shall show how to construct a strong generating set $S$ such that $\mathcal{M}(S)$ is a minimum for $S$ among all strong generating sets for $G$. This construction does not require the computation of any restriction multiplicities.

Assume $\mathcal{K} = \{K_n \geq \cdots \geq K_0\}$. Extend this to a chain $\mathcal{K} \cup \{1\}$ that includes 1, by defining $K_{-1} = 1$. Let $S$ be a subset of $G$. Then $\tilde{S}$ be defined by

$$\tilde{S} = \bigcup_{s \in S} K_{c^+(s)} \cap Centralizer(K_{c^-(s)})$$

where for each $s \in S$, $K_{c^+(s)}$ is the smallest subgroup in $\mathcal{K} \cup \{1\}$ containing $s$, and $K_{c^-(s)}$ is the largest subgroup in $\mathcal{K} \cup \{1\}$ centralizing $s$.

**Lemma 3.19.**
1. *If $S$ is a subset of $T$, then $\gamma(S, \mathcal{K}) \leq \gamma(T, \mathcal{K})$, but $\mathcal{M}(S, \mathcal{K}) \geq \mathcal{M}(T, \mathcal{K})$.*
2. *$S \subseteq \tilde{S}$ and $\mathcal{M}(S, \mathcal{K}) = \mathcal{M}(\tilde{S}, \mathcal{K})$.*

Now we turn to the problem of minimizing $\mathcal{M}(S, \mathcal{K})$, for fixed $\mathcal{K}$, subject to the constraint that the adapted diameter $\gamma(S, \mathcal{K})$ is finite. Assume $\mathcal{K} = \{G =$

$K_n \geq \cdots \geq K_0 = 1\}$. Construct a subset $S_{\mathcal{K}}$ of $G$, inductively as follows. Let $S(0) = K_0 = 1$. Define $S(i) = S_{\mathcal{K}} \cap K_i$ inductively by

$$S(i) = S(i-1) \cup (K_i \cap Centralizer(K_j))$$

where $j$ is maximal with respect to the property that $S(i)$ generates $K_i$. Let $S_{\mathcal{K}} = \cup_{i=0}^{n} S(i)$.

**Theorem 3.20.** *Let all definitions be as aboven and assume that $S_{\mathcal{K}}$ is a strong generating set for $G$, which minimizes $\mathcal{M}(S, \mathcal{K})$ among all strong generating sets relative to the chain $\mathcal{K}$.* I.e., *$S_{\mathcal{K}}$ minimizes $\mathcal{M}(S, \mathcal{K})$ on subsets of $G$ subject to the constraint that $\gamma(S, \mathcal{K})$ is finite.*

It is significant that this construction does not require the evaluation of any restriction multiplicities. The problem of finding the multiplicities, let alone matrix coefficients, is much harder than the computations with group elements required to construct the set $S_{\mathcal{K}}$. Although actually constructing an FFT requires knowledge of the restriction multiplicities and much more, this simple construction of sets with minimal $\mathcal{M}(S)$ greatly speeds up the search for efficient algorithms.

Finally, we note that the construction of $S_{\mathcal{K}}$ allows us to give a definite procedure, albeit a complicated one, for minimizing the product $v(\mathcal{K})\mathcal{M}(S, \mathcal{K})$ over all chains and strong generating sets. It is clear that we need only conduct our search on the set of chains $\mathcal{K}$, as for any chain $\mathcal{M}(S, \mathcal{K})$ will be a minimum at $S = S_{\mathcal{K}}$. Therefore we need only find subgroup chains that minimize $v(\mathcal{K})\mathcal{M}(S_{\mathcal{K}}, \mathcal{K})$. The following lemma shows that we may restrict this search even further.

**Lemma 3.21.** *Assume, $\mathcal{K}, \mathcal{L}$ are subgroup chains containing $G$ and 1, and $\mathcal{L}$ is a refinement of $\mathcal{K}$.*

1. *If $\gamma(S_{\mathcal{K}}, \mathcal{L})$ is finite, then $S_{\mathcal{L}} \subseteq S_{\mathcal{K}}$.*
2. *Any minimum of $v(\mathcal{L})\mathcal{M}(S_{\mathcal{L}}, \mathcal{L})$ occurs at some chain, $\mathcal{L}$ such that $S_{\mathcal{L}}$ is not a strong generating set for any nontrivial refinement $\mathcal{L}'$ of $\mathcal{L}$.*

*Remark 3.22.* It is also possible that we are given a generating set $S$ for $G$, and are asked to construct a subgroup chain so that the complexity of the separation of variables algorithm based on this generating set and subgroup chain is small. If the generating set is minimal, then an ordering of the elements of $S$, as $s_1, \ldots, s_n$ gives rise to a subgroup chain defined by $K_i = \langle s_1, \ldots, s_i \rangle$. A first step to finding a good ordering is to find one that minimizes the numbers $c^+(s_i) - c^-(s_i)$ defined relative to this chain.

One approach to this is to draw a graph with vertices corresponding to the elements of $S$, and edges corresponding to pairs of noncommuting elements. For example, if $G$ is a Weyl group and $S$ is the set of simple reflections, then this graph will be the Coxeter graph for that Weyl group. In general, we have $c^+(s_i) = i$, and $c^-(s_i)$ may be read from the graph as the largest $j$ such that $s_i$ is not connected to any of $s_1, \ldots s_j$ by an edge. In this way we see that finding an ordering that minimizes the numbers $c^+(s_i) - c^-(s_i)$ is related to the problem of drawing the graph in a form that resembles a chain.

3.3. **Separation of variables at the scalar level.** The separation of variables idea can be applied even more fruitfully on the scalar level than on the matrix level. One reason is that by treating the Fourier transform as a collection of scalar transforms, we may better keep track of the products of matrix elements that occur in the separation of variables approach to its computation. Armed with this

information, we can now use the commutativity of scalar multiplication, not present in the matrix case, to change the order of the summations in the algorithm.

The starting point for the scalar approach is the same as in the matrix case. We assume that $G$ is a finite group, $\Delta$ is a representation of $G$, and we have chosen a factorization $g = a_n \cdots a_1$ for each element of $G$. The Fourier transform of $f$ is then given by

$$\hat{f}(\Delta) = \sum_{a_n} \cdots \sum_{a_1} \Delta(a_n) \cdots \Delta(a_1) f(a_n \cdots a_1)$$

As in the matrix algorithm, we want to choose the $a_i$ so that the matrices $\rho(a_i)$ have a block diagonal/block scalar form.

The algorithm will proceed by summing over the $a_i$ in some order, but not necessarily with increasing $i$, and the way that the matrix elements of the matrices factor through the sums may be chosen differently from the matrix case. First, however, we must write the sum (3.3) in coordinates and expand out the matrix multiplications. For this we need a method of matching up the nonzero matrix entries of the factors in this product, and that requires a method of indexing the rows and columns of the matrices that describes the block structure easily. The right combinatorial tool for this last task is the set of paths in a Bratteli diagram, which we shall now define.

3.3.1. *Bratteli diagrams and path configurations.*

**Definition 3.23** (Bratteli diagram)**.** Assume $K_n \geq \cdots \geq K_0 = 1$ is a chain of subgroups. Then the **Bratteli diagram for this chain** is the graded diagram defined as follows.

1. For $i = 0, \ldots, n$, the vertices at level $i$ are the inequivalent irreducible representations of $K_i$. At level 0 there is a unique vertex, 1.
2. If $\alpha$ is at level $i$ and $\beta$ is at level $i-1$, then draw $\mathcal{M}(\alpha, \beta)$ arrows from $\beta$ to $\alpha$, where

$$\mathcal{M}(\alpha, \beta) = \text{ the multiplicity of } \beta \text{ in } \alpha \downarrow K_{i-1}$$

We already have an example of a Bratteli diagram in Section 2. Figure 1 is the Bratteli diagrams for the chain $\mathbf{Z}/8\mathbf{Z} > \mathbf{Z}/4\mathbf{Z} > \mathbf{Z}/2\mathbf{Z} > 1$. The best known Bratteli diagram is the Young lattice (see Figure 2).

Bratteli diagrams have a simple interpretation as an indexing scheme for the block matrices $\rho(a)$ of the preceding section (cf. Lemma 3.9). Assume $\rho$ is an irreducible representation of $K_n$ adapted to the chain $K_n \geq \cdots \geq K_0 = 1$. The arrows from level $n-1$ to the representation $\rho$ at level $n$ index the blocks corresponding to the restriction of $\rho$ from $K_n$ to $K_{n-1}$, with arrows starting at the same vertex indexing identical blocks in different positions along the diagonal. Arrows from level $n-2$ to level $n-1$ index blocks within these blocks, and so on. In this way we can index the blocks for the restriction $K_n \downarrow K_i$ by paths from level $i$ to level $n$. This type of indexing has proved useful in refining the application of the matrix separation of variables approach (cf. [68], esp. Section 4.2).

By restricting all the way down to $K_0 = 1$, we obtain an indexing scheme for the rows and columns, and hence matrix coefficients, of the chain-adapted representation $\rho$. For this, an arrow from an irreducible representation $\gamma$ of $K_i$ to an irreducible representation $\eta$ of $K_{i+1}$ is associated with a $K_i$-equivariant injection of the corresponding representation space $V_\gamma$ into $V_\eta$ such that the images of the injections for different arrows are mutually orthogonal $K_i$-invariant subspaces of

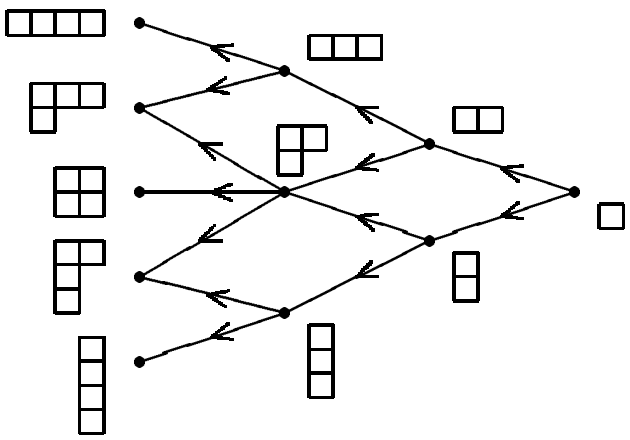


FIGURE 2. The Young lattice for $S_4$.

$V_\eta$. Given any path in the Bratteli diagram from the trivial representation 1 at level 0, to the vertex $\rho$ at level $n$, we may compose all the injections for that path and apply the resulting map to a fixed nonzero vector in the trivial representation. This gives a vector in $V_\rho$. As the path ranges over all possible paths from 1 to $\rho$ in the Bratteli diagram, the set of vectors we obtain makes up a Gel'fand-Tsetlin basis for $V_\rho$ adapted to the chain of subgroups which gave rise to the Bratteli diagram.

*Remark 3.24.* What this does is create an isomorphism between the "path algebra" of the Bratteli diagram and the chain of semisimple algebras defined by the succession of group algebra inclusions $\mathbf{C}[K_i] \hookrightarrow \mathbf{C}[K_{i+1}]$. In this way the group algebra $\mathbf{C}[K_n]$ is realized as a multimatrix algebra (see eg. [44]).

The beauty of the Bratteli diagram formalism lies in the convenient characterization it gives for all types of structured matrices which can arise through the use of Gel'fand-Tsetlin bases.

To begin, consider $a \in K_i \leq K_n$. According to the above explanation, the entries of $\rho(a)$ are indexed by pairs of paths from 1 to $\rho$ in the corresponding Bratteli diagram. Since $a \in K_i$, the matrix entry $\rho_{uv}(a)$ can be nonzero only when paths $u$ and $v$ intersect at the level $i$, *i.e.,* at $K_i$, and agree from level $i$ to level $n$. In this case the matrix coefficient $\rho_{vw}(a)$ is independent of the subpath from level $i$ to $n$. This is precisely the diagrammatic realization of a block diagonal matrix with certain equal sub-blocks.

For another example, consider the situation in which $a \in K_n$ centralizes $K_j$. Using the path algebra formalism, it is not too difficult to show that in this case $\rho_{uv}(a)$ can only be nonzero when $u$ and $v$ agree from level 0 to level $j$, then varying freely until they necessarily meet at $\rho$ at level $n$. Here the matrix coefficient depends only the pairs of paths between levels $j$ and $n$.

These are two of the simplest examples of the relation between index dependence and matrix structure. More complicated structures are easily imagined and easily described.

Matrix products have a simple description in this notation. Suppose $A = (A_{u_1 u_2})$ and $B = (B_{v_1 v_2})$ are block matrices, with rows and columns indexed by paths in a Bratteli diagram, starting at level 0 and ending at $\rho$ on level $n$. To multiply these

matrices, we first form the quantity $A_{u_1u_2}B_{u_2v_2}$, which is indexed by a triple of paths from level 0 to $\rho$. We then sum over all possible values of the path $u_2$. The index sets for all the quantities involved are represented pictorially in Figure 3.

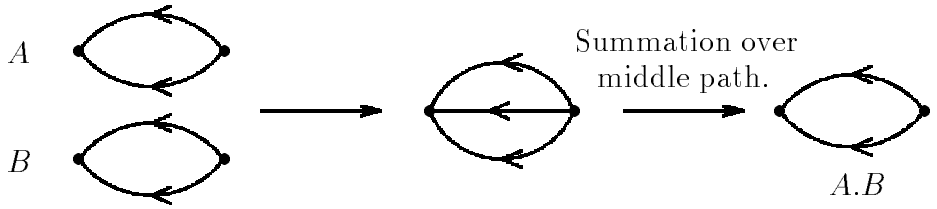



FIGURE 3.

If, instead, $A$ was a block matrix in $\mathrm{span}_{\mathbf{C}}(\rho(K_i))$ and $B$ was in the centralizer of $\rho(K_j)$, then the intermediate quantity in the computation of the matrix product would be indexed by the configuration of paths shown in Figure 4.

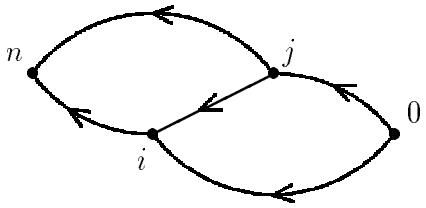



FIGURE 4.

*Remark 3.25.* In the discussion above we have used pairs of paths from level 0 terminating at an irreducible representation $\rho$ to index the matrix elements of $\rho$. If we replace $\rho$ by any representation adapted to the whole chain of subgroups, then we may index matrix elements by pairs of paths in the Bratteli diagram, except that the paths may now end at any irreducible representation at level $n$ which occurs in the representation. As always, the pairs of paths must end at the same vertex for the matrix element to be nonzero. In particular, the nonzero matrix elements of an adapted model representation are indexed by all the pairs of paths from level 0 to level $n$, ending at the same vertex.

3.3.2. *Efficient multiplication of multiple matrix products.* The examples above are simple matrix products. The power of indexing via paths lies in the ability to describe multiple products of structured matrices. In a multiple matrix product, the indexing shows more clearly which matrix entries from which matrices are multiplied, and thereby reveals ways of organizing the computation of the product in ways other than usual matrix multiplication. The idea is that entries of a multiple matrix product are computed as sums of products of matrix entries, and it is possible that certain subsums may occur repeatedly even though these smaller computations are never realized as the result of any complete matrix multiplication. As an easy example, consider the situation in which two matrices $A$ and $B$ have many initial segments of rows equal and columns equal (respectively). Particular entries in the product will require the initial summation of these segments, all of which are the same, none of which gives an actual entry.

**Example 3.26.** Assume the chain of subgroups

$$K_3 \geq K_2 \geq K_1 \geq 1$$

has a partial Bratteli diagram as shown in Figure 5.

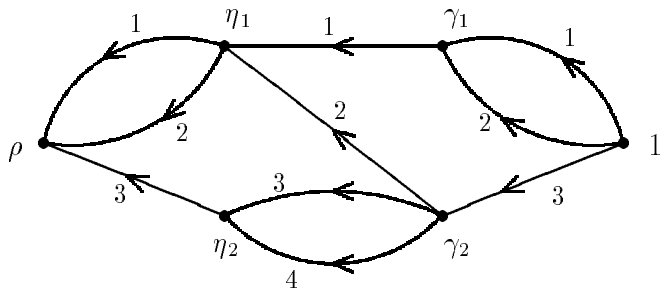



FIGURE 5.

Thus, $\rho$ is an irreducible representation of $K_3$, $\eta_1$ and $\eta_2$ are irreducible representations of $K_2$, and $\gamma_1$ and $\gamma_2$ are irreducible representations of $K_1$, such that

$$\begin{array}{lll} K_1: & d_{\gamma_1} = 2; & d_{\gamma_2} = 1; \\ K_2: & d_{\eta_1} = 3; & d_{\eta_2} = 2; \\ & \eta_1 \downarrow K_1 = \gamma_1 \oplus \gamma_2; & \eta_2 \downarrow K_1 = 2\gamma_1; \\ K_3: & d_\rho = 8; & \\ & \rho \downarrow K_2 = 2\eta_1 \oplus \eta_2; & \rho \downarrow K_1 = 4\gamma_1 \oplus 2\gamma_2. \end{array}$$

Thus, a basis of $V_\rho$ can be indexed by the paths in Figure 5, which in turn are represented as triples $(v_1, v_2, v_3)$ where $v_i$ indexes the path from level $i-1$ to level $i$. For example, $(3, 2, 1)$ indexes the path shown in Figure 6.

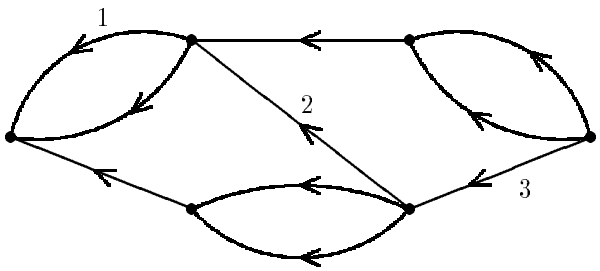



FIGURE 6.

Suppose we would like to compute a product of the form $ABC$ where $A$ and $C$ are in the span of $\rho(K_2)$ and $B$ is in the span of the collection of matrices which centralize the span of $\rho(K_1)$. Simple matrix multiplication says that $w, z$ entry of $ABC$ is

$$(ABC)_{wz} = \sum_x \sum_y A_{wx} B_{xy} C_{yz}$$

where the sums are over all paths $x$ and $y$ in Figure 5. In order to better organize the computation we consider the structure of the matrices $A, B$, and $C$.

As explained above, since $A$ and $C$ are in the span of $\rho(K_2)$ we have that their $x, y$ entries must be zero unless $x$ and $y$ agree on the last leg and furthermore any two paths which agree on the ordered pairs of subpaths given by their first two legs (*i.e.,* subpaths from level 0 to level 2) must be equal. This shows that such matrices must have the following form.

| | 111 | 211 | 321 | 112 | 212 | 322 | 333 | 343 |
|---|---|---|---|---|---|---|---|---|
| 111 | $X_{11}$ | $X_{12}$ | $X_{13}$ | | | | | |
| 211 | $X_{21}$ | $X_{22}$ | $X_{23}$ | | | | | |
| 321 | $X_{31}$ | $X_{32}$ | $X_{33}$ | | | | | |
| 112 | | | | $X_{11}$ | $X_{12}$ | $X_{13}$ | | |
| 212 | | | | $X_{21}$ | $X_{22}$ | $X_{23}$ | | |
| 322 | | | | $X_{31}$ | $X_{32}$ | $X_{33}$ | | |
| 333 | | | | | | | $Y_{11}$ | $Y_{12}$ |
| 343 | | | | | | | $Y_{21}$ | $Y_{22}$ |

Similarly for $B$ in the span of the centralizer of $\rho(K_1)$, the above discussion explains that $B_{v,w}$ must be zero unless $v$ and $w$ agree on the initial leg (*i.e.,* until level 1) and given that, an entry only depends upon the last two steps of its indexing paths in this way, $B$ must have the form

| | 111 | 211 | 321 | 112 | 212 | 322 | 333 | 343 |
|---|---|---|---|---|---|---|---|---|
| 111 | $X_{11}$ | 0 | | $X_{12}$ | 0 | | | |
| 211 | 0 | $X_{11}$ | | 0 | $X_{12}$ | | | |
| 321 | | | $Y_{11}$ | | | $Y_{12}$ | $Y_{13}$ | $Y_{14}$ |
| 112 | $X_{21}$ | 0 | | $X_{22}$ | 0 | | | |
| 212 | 0 | $X_{21}$ | | 0 | $X_{22}$ | | | |
| 322 | | | $Y_{21}$ | | | $Y_{22}$ | $Y_{23}$ | $Y_{24}$ |
| 333 | | | $Y_{31}$ | | | $Y_{32}$ | $Y_{33}$ | $Y_{34}$ |
| 343 | | | $Y_{41}$ | | | $Y_{42}$ | $Y_{43}$ | $Y_{44}$ |

Notice that this matrix has the form

$$(X \otimes I_2) \oplus (Y \otimes I_1)$$

for $X \in M_2(\mathbf{C})$, $Y \in M_4(\mathbf{C})$. Consequently, we see that the full matrix product $ABC$ with entry

$$\sum_{x,y} A_{wx} B_{xy} C_{yz}$$

is in fact a sum over

1. only those $x$'s such that the final leg of $x$ equals the final leg of $w$.
2. only those $y$'s such that the final leg of $y$ equals the final leg of $z$.
3. all pairs of paths $x$ and $y$ must agree on their initial legs (so that $B_{xy}$ is nonzero).

Furthermore,

4. the value of $B_{xy}$ depends only on the final two steps of $x$ and $y$.

Notice that 1, 2 and 4 together imply that the only indices summed on are the middle steps of $x$ and $y$. Thus, we may view the multiplication of $ABC$ in the following schematic fashion: as a sum over the middle paths in subdiagrams of a Bratteli diagram of the form shown in Figure 7.

That is, $w$ and $z$ index the outside paths, and $x_1, x_2, y_2$ are filled in, subject to the constraints depicted in Figure 7. Ordinary matrix multiplication fills in

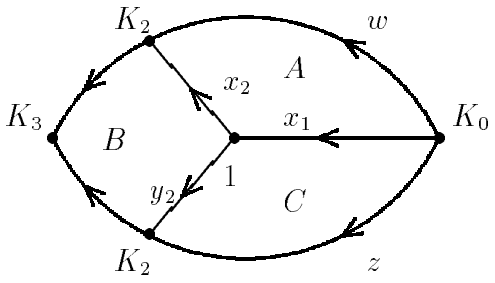



FIGURE 7.

these indices (paths) according to the order of multiplication, but a non-matrix multiplication fills them in according to the values of $x_1$ and then fixing $y_2$ and $x_2$. These are each suggested by the sequence of diagrams shown in Figure 8, where at each step a summation over the dashed paths is indicated.

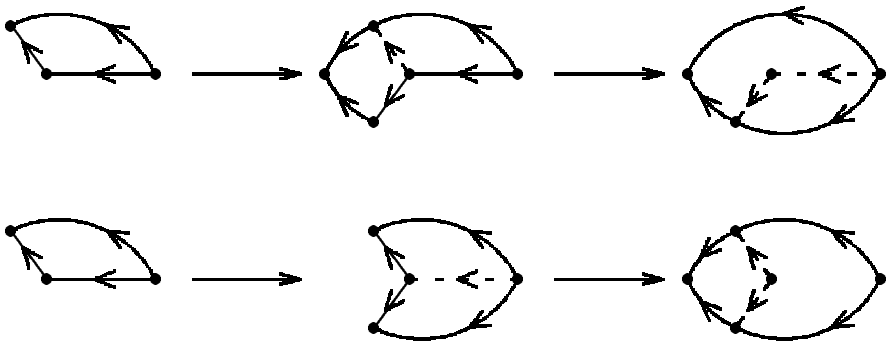


FIGURE 8.

3.3.3. *Contraction and complexity.* The indexing scheme we have described also provides a simple combinatorial method for calculating complexities. We shall only describe this roughly here; full details will appear in the paper [69]. The basic idea is that any collection of matrix multiplications may be described by a configuration of paths, as in the diagrams above, with only some of the paths being summed over. This configuration may be represented by a graded diagram, and the operation of summation gives us a quantity represented by a different graded diagram obtained by removing edges and vertices from the first. Let us call the summation operation **contraction**, as it is a generalization of the notions of contraction of tensors and therefore matrix multiplication. The trick is that contraction only takes as many scalar operations as there are configurations described by the first diagram, and these can be counted.

**Lemma 3.27** (Counting lemma)**.** *For any graded diagram,*

- *label vertices at level $i$ with irreducible representations of $K_i$,*
- *label each edge from a vertex labelled $\alpha$ to a vertex labelled $\beta$ with $\mathcal{M}(\alpha, \beta)$,*

*then the number of scalar operations required to perform any contraction corresponding to the given graded diagram is*

$$\sum_{\substack{\text{labelings}\\ \text{of vertices}}} \prod_{\substack{\text{edges}\\ \alpha \leftarrow \beta}} \mathcal{M}(\alpha, \beta).$$

3.3.4. *Diagrams applied to scalar Fourier transforms.* Using Lemma 3.27 we can obtain the following general result.

**Theorem 3.28.** *Assume $G \geq K$ are finite groups, then the Fourier transform of a function on $G$ can be computed at a complete $K$-adapted set of irreducible representations of $G$ in no more than*

$$\left(2\,|K| + |K \backslash G/K|\, d_K^2\right) |G|$$

*scalar operations, where $d_K$ is the maximum dimension of any irreducible representation of $K$.*

*Proof.* The first step is to write each element of $G$ in the form $g = k_2 a k_1$ where $k_1, k_2$ are elements of $K$ and $a$ comes from a fixed set $A$ of double coset representatives for $K \backslash G/K$. Not all triples $(k_2, a, k_1)$ necessarily occur in our collection of factorizations of group elements. Let $\Delta$ be the model representation of $G$ obtained by summing a complete set of irreducible representations adapted to $G > K$. We start by computing a sum in the tensor product of two spaces of matrices

$$\begin{aligned} F(a) &= \sum_{k_1, k_2} f(k_2 a k_1) \Delta(k_2) \otimes \Delta(k_1) \\ &= \sum_{k_2} \Delta(k_2) \otimes \sum_{k_1} \Delta(k_1) f(k_2 a k_1) \end{aligned}$$

for each $a$ in $A$. This sum may be computed by summing on $k_1$ first and then on $k_2$, and takes a total of $2\,|K|\,|G|$ scalar operations. If $N$ denotes the order of $G$, then $F(a)$ lies in $\mathbf{R}^N \otimes (\mathbf{R}^N)^* \otimes \mathbf{R}^N \otimes (\mathbf{R}^N)^*$, and $\rho(a)$ lies in $\mathbf{R}^N \otimes (\mathbf{R}^N)^*$. For each $a$ we must contract the middle $(\mathbf{R}^N)^* \otimes \mathbf{R}^N$ indices of the tensor $F(a)$ against $\rho(a)$ which lies in $\mathbf{R}^N \otimes (\mathbf{R}^N)^*$. Now we make use of the special structure of the tensor $F(a)$, and the matrix $\Delta(a)$. Their entries may be indexed by configurations of paths in the Bratteli diagram of the chain $G > K$. The intermediate quantity we must compute before performing the summations for the contractions is indexed by the configuration of paths shown in Figure 9, where representations of $G$ are at level 2 and representations of $K$ are at level 1. A simple application of Lemma 3.27

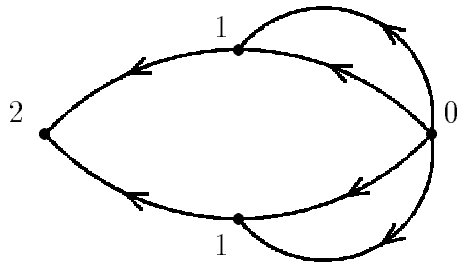



FIGURE 9.

shows that the number of operations used to perform the contraction is

$$\sum_{\substack{\gamma\in\hat{G}\\ \alpha,\beta\in\hat{K}}} \mathcal{M}(\gamma,\alpha)\mathcal{M}(\gamma,\beta)d_\alpha^2 d_\beta^2.$$

It is easy to see that this is bounded by $\sum_{\rho\in\hat{G}} d_\gamma^2 = |G|$, so the total number of operations required to perform all the contractions and sum them together is $|K\backslash G/K|\cdot|G|$. Adding this to the operations count for the computation of all the $F(a)$ gives the desired result. □

This result can be improved by noticing that the first two steps in the computation, summing over the subgroup, are actually Fourier transforms on the subgroup, so FFT techniques may be applied to these sums as well. It is also conceivable that the final sum over double coset representatives could be simplified, but this would presumably require a better understanding of the matrix coefficients as "special functions" of $a$.

**Example 3.29.** If $G = SL_2(q)$ is the group of unimodular $2\times 2$ matrices over a finite field with $q$ elements, then a convenient subgroup to which we may apply Theorem 3.28 is the subgroup $N$ of all unipotent upper triangular matrices.

**Corollary 3.30.** *The Fourier transform of a complex function on $SL_2(q)$ may be computed at a complete set of irreducible representations adapted to the chain $SL_2 > N$, in $(4q-2)\cdot|SL_2|$ scalar operations.*

This corollary improves on the results of [62]. As another example, Theorem 3.28 can be applied to the symmetric groups of low order, adapting to a well-chosen abelian subgroup.

A similar technique may be applied to the general linear group over a finite field, $GL_n(q)$. The factorization used there is a analogous to the generalized Euler angle decomposition for classical compact groups that we shall consider in the next section. In fact the proof of that case follows the derivation of fast algorithms for compact groups very closely, so we shall simply state the result here.

**Theorem 3.31.** *For any $n \geq 2$ and $q \geq 2$, the Fourier transform of a complex function on $GL_n(q)$ may be computed at a complete set of irreducible representations adapted to the chain*

$$GL_n > GL_{n-1} > GL_{n-2} > \cdots > GL_1$$

*in no more than*

$$4^{n-2}(1+\frac{4}{q})q^n\,|GL_n(q)|$$

*scalar operations.*

The scalar separation of variables technique also works particularly well on homogeneous spaces where the associated spherical functions may sometimes be factorized in terms of functions defined on subsets of $G$.

*Remark 3.32.* All the techniques we have described for computing FFTs on finite groups are really using the fact that the group algebra $\mathbf{C}[G]$ is a semisimple algebra. It is therefore possible to state analogous results in any semisimple algebra.

3.4. **Cyclic groups of prime order, and other abelian groups.** We now describe some of the methods for computing Fourier transforms on finite groups that are not solely based on a separation of variables argument.

So far, all the algorithms we have considered for FFTs over finite groups rely in some way on the use of subgroups. Thus, the question arises of what to do when there are no subgroups? The only situation in which this can occur is that in which the group is cyclic of prime order, *i.e.,* it is equal to $\mathbf{Z}/p\mathbf{Z}$ for $p$ a prime. In this case the method of Rader [78] allows us to relate the problem to a convolution on a cyclic group of order $p-1$. Alternatively, the chirp-$z$ transform due to Rabiner et al. [77] allows us to relate the problem to a cyclic group of higher order. These methods, in conjunction with the Cooley-Tukey algorithm, provides an $O(n \log n)$ algorithm for any $n$. They may be explained as follows.

We first need to recall the connection between the DFT and convolution over the group $\mathbf{Z}/N\mathbf{Z}$. If $f$ and $h$ are function on $\mathbf{Z}/N\mathbf{Z}$, then the **convolution** of $f$ and $h$, is a function $f * h$ on $\mathbf{Z}/N\mathbf{Z}$ defined by

$$f * h(x) = \sum_{y=0}^{N-1} f(x-y)h(y). \tag{3.10}$$

Notice that convolution is nothing more than multiplication in the group algebra. Convolution and the Fourier transform enjoy the following useful relationship,

$$\widehat{f * h}(j) = \hat{f}(j)\hat{h}(j). \tag{3.11}$$

As a consequence, the cyclic FFT allows the convolution of two functions $f, h$ on $\mathbf{Z}/N\mathbf{Z}$ to be computed in efficiently via the algorithm

1. Compute $\hat{f}$ and $\hat{h}$.
2. Compute the pairwise products $\hat{f}(j)\hat{h}(j)$ for each $j$.
3. Compute the inverse Fourier transform of the sequence,

$$\hat{f}(0)\hat{h}(0), \ \ldots \ , \hat{f}(N-1)\hat{h}(N-1).$$

This can be accomplished via a standard DFT.

This algorithm uses three FFTs and one pointwise multiplication, instead of the $n^2$ operations needed for direct evaluation of the convolution. Rader's FFT for cyclic groups of prime order is obtained by relating the DFT computation to a convolution algorithm.

To explain, let $N = p$ be a prime. Then $\mathbf{Z}/p\mathbf{Z}$ is a field under the usual addition and multiplication modulo $p$, and the multiplicative group of nonzero elements $(\mathbf{Z}/p\mathbf{Z})^\times$ is a cyclic group of order $p-1$. Fix a generator $g$ of $(\mathbf{Z}/p\mathbf{Z})^\times$, and let $\omega = e^{2\pi i/p}$. Making the substitutions $j = g^a$, $k = g^{-b}$ for the nonzero elements of $\mathbf{Z}/p\mathbf{Z}$ rewrites the Fourier transform of $f$ on $\mathbf{Z}/p\mathbf{Z}$ as

$$\hat{f}(g^{-b}) = \hat{f}(0) + \sum_{a=0}^{p-2} f(g^a)\omega^{g^{a-b}}. \tag{3.12}$$

The summation in (3.12) is the convolution on $\mathbf{Z}/(p-1)\mathbf{Z}$, of the sequence $f(g^a)$, with the sequence $\omega^{g^a}$, and may therefore be computed using Fourier transforms of length $p-1$. The sequence $\omega^{g^a}$ is input independent, so we only need two such Fourier transforms. Thus, $T(p)$, the number of operations needed to compute a

DFT for $\mathbf{Z}/p\mathbf{Z}$, satisfies the bound,

$$T(p) \leq 2 \cdot T(p-1) + 3p - 3. \tag{3.13}$$

If $p$ is a prime bigger than 3, then $p-1$ is composite and $\mathbf{Z}/(p-1)\mathbf{Z}$ has nontrivial proper subgroups. We may therefore use the Cooley-Tukey algorithm to compute Fourier transforms on $\mathbf{Z}/(p-1)\mathbf{Z}$.

The idea of writing the Fourier transform in terms of a convolution does not depend on $N$ being prime. Rabiner et al. [77] (see also [13]) make the change of variables $jk = (j^2 + k^2 - (j-k)^2)/2$ to obtain

$$\hat{f}(k) = \omega^{k^2/2} \sum_{j=0}^{N-1} \left( f(j)\omega^{j^2/2} \right) \omega^{(j-k)^2/2}$$

This is a non-cyclic convolution of the sequence $\left( f(j)\omega^{j^2/2} \right)$ with the sequence $\left( \omega^{-j^2/2} \right)$, and may be performed using a cyclic convolution of any length $M \geq 2N$. This gives us the bound

$$T(N) \leq 2 \cdot T(M) + M + N - 1. \tag{3.14}$$

Choosing $M$ to be the smallest power of 2 greater than $2N$ immediately gives an $O(N \log N)$ algorithm, though better choices are possible. This method is commonly known as the chirp-$z$ transform.

We have seen three distinct methods for computing Fourier transforms on cyclic groups. This raises the question: Which combination of methods is best for a given group $\mathbf{Z}/N\mathbf{Z}$? Diaconis [30] gives a analysis of the average running times of algorithms based on each of the methods. Baum, Clausen, and Tietz [10] take a careful look at the choice of the number $M$ occurring in the chirp-$z$ approach, and prove a good bound valid for any cyclic group. Combining this with the result on the Fourier transform of a direct product of groups (see Section 3.5), they obtain the following theorem.

**Theorem 3.33** (Baum, Clausen, Tietz [10]). *The Fourier transform of a function on any finite abelian group $G$ may be computed in less than $8\,|G| \log |G|$ scalar operations.*

3.5. **Tensor Products and Group Extensions.** We have already seen that we can relate the Fourier transform on a finite group $G$ to Fourier transforms on a subgroup, $K$. If the subgroup is normal, then it is natural to ask whether we can also use Fourier transforms on the quotient group, $G/K$ to give more efficient algorithms. One trouble with this idea is that there is no obvious way of constructing a representation on $G/K$ from a representation on $G$. However we can take the tensor product of a representation of $G/N$ with a representation of $G$ to get a new representation on $G$. This suggests an approach to the computation of Fourier transforms based on factorization in the space of representations rather than factorization of group elements. We have already seen such methods work in Yates' algorithm for the Walsh-Hadamard transform.

The works of Beth on solvable groups [12], Clausen and Baum [19] on solvable and supersolvable [7] groups and Rockmore on abelian group extensions [83] are based on the idea of factoring both the group elements and the representations. In this respect, their algorithms differ from a pure separation of variables algorithm. These methods are all based on versions of the following general approach.

Assume $G$ is a finite group, $N$ is a subgroup of $G$, $Y$ is a subset of $G$, and $\mathcal{R}$ is a set of matrix representations of $G$. Then let $M_N(Y,\mathcal{R})$ denote the minimum number of operations required to compute the matrix products $\rho(y)\cdot F(y,\rho)$, for all $y\in Y$ and $\rho\in\mathcal{R}$, where the matrices $F(y,\rho)$ are in the span of $\rho(N)$. Define

$$m_N(Y,\mathcal{R}) = \frac{1}{|G|} M_N(Y,\mathcal{R}).$$

**Theorem 3.34.** *Assume $G$ is a finite group, and $N$ is a normal subgroup of $G$. Let $\mathcal{S}$ be any set of representations of $G/N$, and let $\mathcal{R}$ be a set of representations of $G$ adapted to $G$ and $N$. Let $Y$ be a complete set of coset representatives for $G/N$. Then*

$$t_G(\mathcal{S}\otimes\mathcal{R}) \le t_N(\mathcal{R}_N) + \left(\frac{1}{|N|}\sum_{\rho\in\mathcal{R}} d_\rho^2\right) t_{G/N}(\mathcal{S}) + m_N(Y,\mathcal{R})$$

*Proof.* Assume $f$ is a complex function on $G$. Let $\sigma$ be a representation in $\mathcal{S}$ and let $\rho$ be a representation in $\mathcal{R}$. For each $y\in Y$, we define a function on $N$, by $f_y(n) = f(yn)$ for any $n\in N$.

$$\hat{f}(\sigma\otimes\rho) = \sum_{y\in Y} \sigma(y)\otimes\left[\rho(y)\hat{f_y}(\rho\downarrow N)\right]$$

The algorithm now proceeds in three stages. First we compute all the Fourier transforms of the functions $f_y$ on $N$, at the set of representations $\mathcal{R}_N$, and use these to build the transforms $\hat{f}(\rho\downarrow N)$. This takes $|G|\,t_N(\mathcal{R}_N)$ scalar operations.

In the second stage, we compute the matrix products $\rho(y)\cdot\hat{f_y}(\rho\downarrow N)$ for all $y\in Y$ and $\rho\in\mathcal{R}$. This requires $|G|\,m_N(Y,\mathcal{R})$ scalar operations, by definition.

Finally we note that the remaining sum of tensor products, can be viewed as a collection of $\sum_{\rho\in\mathcal{R}} d_\rho^2$ scalar Fourier transforms on $G/N$, at each of the representations in $\mathcal{S}$. These transforms clearly take a total of $(\sum_{\rho\in\mathcal{R}} d_\rho^2)\cdot(|G|/|N|)\cdot t_{G/N}(\mathcal{S})$ scalar operations. □

Applying Theorem 3.34 to a direct product of groups $H\times K$, immediately gives the usual result [19, 23, 12, 68] for the complexity of the Fourier transform on a group product. Using sets of matrix representations adapted to both $H$ and $K$ simultaneously, we have $t_{H\times K}\le t_H + t_K$.

3.5.1. *Abelian group extensions and solvable groups.* We now describe how Theorem 3.34 can be applied to an abelian group extension.

Assume $G$ is a finite group, and $N$ is a normal subgroup of $G$ such that $G/N$ is abelian. Then the irreducible representations of $G/N$ are irreducible and form a group which acts on the irreducible representations of $G$ via the tensor product. If we let $\mathcal{R}_0$ be an $N$-adapted set of representatives for the orbits under this action, then $\widehat{G/N}\otimes\mathcal{R}_0$ contains a complete set of irreducible representations for $G$. Applying Theorem 3.34 to this set of representations gives a method of computing Fourier transforms on $G$.

Beth, in his work on solvable groups [12] (see also [19]), and Rockmore, in his work on abelian group extensions [83], use two additional ideas in conjunction with the method of Theorem 3.34:

1. The action of $\widehat{G/N}$ on equivalence classes of irreducible representations of $G$ is not, in general, free. To make efficient use of Theorem 3.34, we must apply it to

the representations orbit by orbit. For each orbit generated by a representation $\rho$ in $\mathcal{R}$, there is a subgroup $H_\rho$ containing $N$, such that any representation in the orbit may be obtained by extending a representation of $N$ to $H_\rho$ and then inducing up to $G$. We use the separation of variables technique to reduce the computation to a collection of Fourier transforms at representations of $H_\rho$ followed by a sum over coset representatives of $G/H_\rho$ of matrix products involving these transforms. Let $\mathcal{R}_\rho$ denote the set of representations of $H_\rho$ that occur in this procedure. Then $\widehat{H_\rho/N}$ acts on $\mathcal{R}_\rho$ via tensor products and this action is free. Therefore, we may use Theorem 3.34 to calculate the transforms on $H_\rho$ at the representations of $\mathcal{R}_\rho$ efficiently.

2. Recall that a matrix is called **monomial** if it has at most one nonzero entry in each row or column. A matrix is called **block monomial** if it is a block matrix, with at most one nonzero block intersecting each row or column.

**Lemma 3.35.** *Any irreducible representation $\rho$ of a finite group $G$ which is induced from a representation on a normal subgroup $H$ and which is also adapted to that subgroup, must have block monomial form, according to blocks occurring in its restriction to $H$. In addition, as $g$ ranges over a set of coset representatives for $G/H$, the nonzero blocks in the different matrices $\rho(g)$ never coincide.*

We now apply Lemma 3.35 to the current situation. The representation $\rho$ is induced from $\rho_{H_\rho}$, so we require it to be adapted to $H_\rho$. In this case, the representation matrices $\rho(g)$ for $g \in G$ are $d_\rho \times d_\rho$ block matrices with blocks of size $d_\rho/[G : H_\rho]$, and at most one block intersects each row or column of $\rho(g)$. Multiplication of a block monomial matrix by a block diagonal matrix can be done much faster than the multiplication of two arbitrary matrices. This can be used to speed up the matrix multiplications that come from the use of separation of variables. Also note that summation of block monomial matrices of this form over coset representatives of $G/H_\rho$ does not require any arithmetic operations.

Putting these ideas together gives the following Theorem.

**Theorem 3.36.** *Assume $G$ is a finite group, and $N$ is a normal subgroup of $G$ such that $G/N$ is abelian. Let $\mathcal{R}_0$ be a complete set of representatives for the orbits of $\widehat{G/N}$ acting on irreducible representations of $G$ via tensor products. Let $\mathcal{R}$ be a complete set of irreducible matrix representations of $G$ constructed from those in $\mathcal{R}_0$ by taking tensor products with representations in $\widehat{G/N}$. Assume that $\mathcal{R}_0$ is adapted to $N$, and each $\rho \in \mathcal{R}_0$ is adapted to $H_\rho$. Then*

$$\begin{aligned} t_G(\mathcal{R}) &\leq t_N(\mathcal{R}_N) + \frac{1}{|G|}\sum_{\rho\in\mathcal{R}} \frac{|H_\rho|}{|G|} d_\rho^2 t_{H_\rho/N} + \frac{2}{|G|}\sum_{\rho\in\mathcal{R}} \frac{|H_\rho|}{|G|} d_\rho^3 \\ &\leq t_N(\mathcal{R}_N) + \max_{\rho\in\mathcal{R}_0}\{t_{H_\rho/N}(\widehat{H_\rho/N})\} + 2\,|N|^{\frac{1}{2}} \end{aligned}$$

*If $G/N$ is cyclic of prime order, then*

$$t_G(\mathcal{R}) \leq t_N(\mathcal{R}_N) + t_{G/N}(\widehat{G/N}) + |N|^{\frac{1}{2}}$$

*Remark 3.37.* There are several variations of this result that may be obtained by performing the matrix multiplications before the transforms on $H_\rho/N$, or by applying Theorem 3.34 to a different subgroup altogether.

One important way in which Theorem 3.36 differs from other results in this paper is that it assumes adaptability of different representations to subgroups that may not form a chain. However, if $G/N$ is cyclic of prime order, then the subgroups $H_\rho$ must either be $G$ or $N$. It follows that we may get a result for representations adapted to a composition series for any solvable group by applying this result recursively to cyclic extensions of prime index.

**Theorem 3.38** (Beth, Clausen)**.** *Assume $G_n > \cdots > G_0$ is a composition series of the finite solvable group, $G$. Then the number of scalar operations required to compute the Fourier transform of a function on $G$ at a complete chain-adapted set of irreducible representations of $G$ is no more than*

$$\frac{7}{2\sqrt{[G_n : G_{n-1}]}}\,|G|^{\frac{3}{2}} + 8\,|G|\log G.$$

3.5.2. *Supersolvable groups.* Theorems 3.36 and 3.38 depend on the use of representations whose matrices are block monomial. Baum [7] has shown that for a large class of groups, the supersolvable groups, the representation matrices in a basis adapted to a suitable chain of subgroups are actually monomial. Putting this together with Theorem 3.36 gives a fast transform in $O(|G|\log|G|)$ operations, on any supersolvable group, $G$.

**Definition 3.39.** A **chief series** for the group $G$ is a chain of subgroups,

$$G = G_n > \cdots > G_0 = 1$$

of $G$, such that each $G_i$ is a normal subgroup of $G$, and $G_{i-1}$ is maximal among normal subgroups of $G$ properly contained in $G_i$. A supersolvable group is a group with a chief series whose successive quotients $G_i/G_{i-1}$ are cyclic of prime order.

**Lemma 3.40** (Baum)**.** *Assume $G$ is a supersolvable group, and $\rho$ is a representation of $G$ adapted to a chief series for $G$. Then for any $g \in G$, the representation matrix $\rho(g)$ is monomial.*

This result shows us that if we apply the algorithms of Theorem 3.36 to a representations of a supersolvable group, adapted to a chief series, then the only matrix multiplications that occur have one of the factors monomial. Baum used this observation to derive the following theorem.

**Theorem 3.41** (Baum)**.** *Assume $G_n > \cdots > G_0$ is a chief series of the finite supersolvable group, $G$. Then the number of scalar operations required to compute the Fourier transform of a function on $G$ at a complete chain-adapted set of irreducible representations of $G$ is no more than*

$$(8.5)\,|G|\log|G|.$$

*Remark 3.42.* The algorithms we have seen for computing Fourier transforms of functions on finite groups, usually require prior knowledge of the representation matrices on a set of generators for that group. In general, constructing these matrices is a difficult problem, though a general purpose algorithm for doing this in polynomial has been developed by Babai and Rónyai [5].

In the paper [8], Baum and Clausen show that for supersolvable groups, this problem may be efficiently solved. More specifically, given a supersolvable group specified by a power-commutator presentation, they show how to construct the representation matrices, for a generating set, at a complete set of representations

adapted to a chief series. Their algorithm is extremely efficient. They show that it only requires $14\,|G|\log|G| + 5\,|G|$ "basic" operations, where a basic operation is either an arithmetic operation in a certain cyclic group, a multiplication of permutations in a symmetric group, or copying an entry of a monomial matrix.

## 4. FFTs for compact Lie groups

The generalizations of the fast Fourier transforms that we have pursued so far view the Fourier transform algorithms of Yates and Cooley as finite group phenomena. However, most Fourier analysis used in practice comes from continuous spaces. That is, the natural phenomena under study are continuous, but finite discretization is necessary to perform calculations.

The classical example is the case of the circle. This has already been discussed in some detail in Section 2.2. Recall that in that section, the Cooley-Tukey FFT was cast as both an algorithm for finite groups as well as one for efficiently computing the Fourier transform of a so-called bandlimited function on the continuous compact abelian group of the circle. It is the latter formulation which is of interest for this section.

Stated in this way, it is natural to ask if such an algorithm might generalize to nonabelian continuous compact groups. The first real breakthrough in this direction occurred in 1989 and was due to Driscoll and Healy [33]. Motivated by potential applications in a variety of areas, including computer vision and meteorology, they described an efficient exact algorithm for computing the Fourier transform of a bandlimited function defined on the two-sphere. Previous to their work all other efficient algorithms developed (also in the context of meteorological applications) were only approximate in nature [75, 2].

To state things a bit more precisely, recall that any integrable function on the two-sphere has an expansion (its Fourier expansion) in terms of spherical harmonics. The spherical harmonics are to the two-sphere as the complex exponentials are to the circle – for each integer $l \geq 0$, the spherical harmonics of order $l$ span a group-invariant irreducible subspace of functions, the collection of which describe a basis for the integrable functions on the two-sphere. In this case, the group is the (noncommutative) matrix group $SO(3)$. A bandlimited function of bandlimit $b$ is simply any function in the span of the harmonics of order less than $b$, for some $b \geq 0$, in which case it has at most $b^2$ Fourier coefficients. Driscoll and Healy produced an algorithm which in exact arithmetic, computes exactly the Fourier coefficients of a bandlimited function of bandlimit $b$ from a collection of $2b^2$ samples in $O(b^2 \log^2 b)$ operations. This is in contrast to the $b^4$ operations that a direct computation would require and the $b^3$ operations that a separation of variables approach requires. The key algorithmic advance in Driscoll and Healy's work is a fast algorithm for computing associated Legendre transforms, for which they give an $O(b \log^2 b)$ algorithm, versus the $b^2$ algorithm that a direct computation takes. The a priori error analysis in [33] as well as subsequent numerical experiments and algorithmic improvements [49] strongly suggest the possibility of effective and reliable implementation of this algorithm. The advent of such an algorithm has made possible the use of spectral methods for solving PDEs in spherical geometry, a technique which was often avoided due to the expense of direct computation of Legendre transforms [84].

Fast Fourier transforms on more general compact groups are a very natural setting for the separation of variables technique. For the classical groups, the factorization of group elements using generalized Euler angles gives a coordinate system which readily suggests the separation of variables technique, and the idea of using Gel'fand-Tsetlin bases, Schur's Lemma, and commutativity to obtain representation matrices of a special structure has a long history in this context. Indeed, Gel'fand-Tsetlin bases, and adapted representations were originally invented for the calculation of matrix coefficients, using these methods [43]. Since then there has been a tremendous amount of work done, much of it centered around N. J. Vilenkin and A. U. Klimyk, finding explicit expressions for the matrix coefficients [42, 43, 58, 60, 61, 91]. Thus most of the ideas needed to develop fast transforms on compact Lie groups and their homogeneous spaces have existed for a long time. The main new idea, as for the sphere, has been the introduction of fast polynomial transforms.

One nice feature of the separation of variables technique for compact groups is that any abstract results formulated in this setting also apply to finite groups. In this way it can be shown that an abstract formulation of the Euler angle decomposition for $SO(n)$ can be used to derive Clausen's algorithm for the symmetric group, and at the same time we get the new FFTs for the general linear group that we discussed in the previous section. On the other hand, the higher dimensional real, complex and quaternionic spheres may be treated in a fashion almost identical to the Driscoll-Healy treatment of $S^2$. Thus the compact Lie group case provides a bridge between the finite group and fast polynomial transforms. We now turn to a brief survey of these questions following the lines of [65, 66, 64].

4.1. **Example: Fast Fourier analysis on the rotation group.** The Fourier transform on $SO(3)$ provides a good paradigm for the use separation of variables techniques on both compact Lie groups and finite groups. It is one of the few group examples where the techniques based on factorization of group elements may be supplemented by the fast polynomial transform techniques of Driscoll and Healy to produce an even faster transform. Other cases where this occurs are usually homogeneous spaces.

4.1.1. *The Fourier transform on the rotation group.* As usual, we start by factoring elements of $SO(3)$. Define matrices,

$$r_2(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad r_3(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}. \tag{4.1}$$

Then any element, $g \in SO(3)$ may be expressed uniquely in the form

$$g = r_2(\varphi_2) r_3(\theta) r_2(\varphi_1) \tag{4.2}$$

for $\varphi_1, \varphi_2 \in [0, 2\pi)$ and $\theta \in (0, \pi)$ or $\varphi_1 = \varphi_2 = 0$ and $\theta \in \{0, \pi\}$. This defines a coordinate system on a dense open subset of the rotation group, and the coordinates, $\varphi_2, \theta, \varphi_1$, of any rotation are called its **Euler angles**.

*Remark 4.1.* The set of group elements $\{r_3(\theta) | \theta \in [0, \pi]\}$ is actually a set of double coset representatives for $SO(3)$ relative to the subgroup $SO(2)$, so the factorization (4.2) is a continuous analogue of the factorization used in the proof of Theorem 3.28. Likewise, the method we shall use to compute Fourier transforms on $SO(3)$ is a continuous analogue of the methods used in Theorem 3.28.

It is well-known (see e.g. [91]) that the representations of $SO(3)$ are indexed by the nonnegative integers, such that that the $l^{\text{th}}$ representation $V_l$ has dimension $2l+1$. We index the matrix coefficients of the representation $V_l$ by a pair of integers, $m_1, m_2$ satisfying $|m_1|, |m_2| \leq l$. Then, in a suitable basis, the matrix coefficients of $V_l$ may be expressed as functions of the Euler angles of the form

$$T^l_{m_1,m_2}(\theta, \varphi_1, \varphi_2) = P^l_{m_1,m_2}(\theta) e^{im_1\varphi_1} e^{im_2\varphi_2} \tag{4.3}$$

where $P^l_{m_1,m_2}$ is written in terms of Jacobi polynomials as

$$P^l_{m_1,m_2}(\theta) = \left[\frac{(l-m_1)!(l+m_1)!}{(l-m_2)!(l+m_2)!}\right]^{\frac{1}{2}} \times \\ \times (\sin\frac{\theta}{2})^{m_1-m_2}(\cos\frac{\theta}{2})^{m_1+m_2} P^{(m_1-m_2,m_1+m_2)}_{l-m_1}(\cos\theta)$$

Both the product form of (4.3) and the way the indices are grouped among the three factors follow directly from the use of a Gel'fand-Tsetlin basis, i.e. a basis that respects the decomposition of the restriction of $V_l$ to the subgroup $SO(2)$. For a proof of this formula, see [91] or [60].

**Definition 4.2** (Fourier transform on $SO(3)$)**.** Assume $f$ is a continuous complex-valued function on $SO(3)$. Then the **Fourier transform of** $f$ is the collection of numbers $\hat{f}(V_l)(m_1, m_2)$ defined by

$$\hat{f}(V_l)_{m_1,m_2} = \frac{1}{8\pi^2}\int_{S^2} f(\varphi_2, \theta, \varphi_1) P^l_{m_1,m_2}(\theta) e^{i(m_1\varphi_1 + m_2\varphi_2)} \sin\theta d\varphi_1 d\varphi_2 d\theta \tag{4.4}$$

By the Peter-Weyl theorem for the group $SO(3)$, the collection of matrix coefficients (4.3) forms an orthogonal basis for the space of square integrable functions on $SO(3)$. If $f$ is a continuous function on $SO(3)$, then $(2l+1)\hat{f}(V_l)_{-m_1,-m_2}$ is the coefficient of the $T^l_{m_1,m_2}$ in this expansion.

4.1.2. *Sampling on the rotation group.* In order to make sense of the problem of computing the Fourier transform of a function on $SO(3)$, we still need to relate this problem to one involving finite sums. In practice this means we first need to sample the function whose Fourier transform we wish to compute, at a finite set of points $X$, and then approximate the Fourier transform by sums of the form

$$\sum_{x\in X} w(x) f(x) T^l_{m_1,m_2}(x) \tag{4.5}$$

where $w(x)$ is a weighting function. To make effective use of the factorization (4.2), we shall assume that the set $X$ is a grid in the coordinate system determined by this factorization. For the purposes of the current exposition we shall use the following sets $X_b$, defined for any positive integer $b$ by

$$X_b = \{r_2(\varphi_2^{j_2}) r_3(\theta^k) r_2(\varphi_1^{j_1}) | 0 \leq j_1, j_2, k \leq 2b\}$$

where

$$\varphi_1^j = \varphi_2^j = \frac{j}{2b+1} \quad \text{and} \quad \theta^k = \frac{(k+\frac{1}{2})\pi}{2b+1}.$$

In general, the sum (4.5) does not compute the Fourier transform of the function $f$, but instead, computes the Fourier transform of a measure constructed from $f$.

To ensure that we can still get the Fourier transform of $f$ from these sums, we need to place some conditions on $f$.

**Definition 4.3** (Band-limited functions on $SO(3)$)**.** A continuous function $f$ on $SO(3)$ is **band-limited with band-limit** $b$ if $\hat{f}(V_l)_{m_1,m_2} = 0$ for all $l > b$.

For functions of band-limit $b$, we can choose a weight function for the set $X_b$ such that (4.5) exactly computes the Fourier transform of a band-limited function of band-limit $b$, when $l \leq b$. For this, let $w_b(k)$, $0 \leq k \leq 2b$, be the unique solution to the system of linear equations

$$\sum_{k=0}^{2b} w_b(k) \cdot P_m(\cos\theta^k) = \delta_{0m} \text{ for } 0 \leq m \leq b, \tag{4.6}$$

where $P_m$ is the Legendre polynomial of degree $m$ on $[-1, 1]$. The weighting function we will use for $X_b$ is defined by $w(j_2, k, j_1) = (1/(2b+1)^2)w_b(k)$.

*Remark 4.4.* The Legendre polynomials which appear in equation (4.6) are the spherical functions for the double coset space $SO(2)\backslash SO(3)/SO(2)$. The construction of the weight function $w_b$ by using the zonal spherical functions is generalized in Lemma 4.14.

**Theorem 4.5.** *Assume $f$ is a band-limited function of band-limit $b$. Then for $l \leq b$*

$$\hat{f}(V_l)_{m_1,m_2} = \frac{1}{(2b+1)^2} \sum_{k=0}^{2b} w_b(k) P^l_{m_1,m_2}(\theta^k) \times \\ \times \left[ \sum_{j_2=0}^{2b} e^{im_2\varphi_2^{j_2}} \left[ \sum_{j_1=0}^{2b} e^{im_1\varphi_1^{j_1}} f(\varphi_2^{j_2}, \theta^k, \varphi_1^{j_1}) \right] \right]. \tag{4.7}$$

*Proof.* The rule for tensoring two representation of $SO(3)$ is the famous Clebsch-Gordan formula (see e.g. [60])

$$V_{l_1} \otimes V_{l_2} \cong \bigoplus_{l=|l_1-l_2|}^{l_1+l_2} V_l.$$

From this it follows that the product of two matrix coefficients $T^{l_1}_{m_1,m_2} \cdot T^{l_2}_{n_1,n_2}$ may be written as a linear combination of matrix coefficients of representations $V_l$ with $|l_1 - l_2| \leq l \leq l_1 + l_2$.

Now assume $f$ is a band-limited function of band-limit $b$ and $l \leq b$. Then the definition of the Fourier transform (4.4) may be expanded as the integral of linear combination of products of matrix coefficients coming from representations $V_{l_1}, V_{l_2}$, with $l_1, l_2 \leq b$. As just noted, the Clebsch-Gordan formula implies that this may be written as the integral of a linear combination of matrix coefficients coming from representation $V_s$, with $s \leq 2b$. Therefore it suffices to show that the weighted sums defined by $X_b$ and $w(j_2, k, j_1)$ integrate exactly any matrix coefficient coming from

a representation $V_s$ with $s \leq 2b$. This is easy, as for $|m_1|, |m_1| \leq s \leq 2b$,

$$\frac{1}{(2b+1)^2} \sum_{j_1,j_2,k=0}^{2b} w_b(k) P^l_{m_1,m_2}(\theta^k) e^{i(m_1\varphi^1_{j_1} + m_2\varphi^{j_2}_2)}$$

$$= \delta_{0,m_1}\delta_{0,m_2} \sum_{k=0}^{2b} w_b(k) P^l_{m_1,m_2}(\theta^k)$$

$$= \delta_{0,m_1}\delta_{0,m_2} \sum_{k=0}^{2b} w_b(k) P^l_{0,0}(\theta^k) = \delta_{0,m_1}\delta_{0,m_2}\delta_{0,k}$$

□

4.1.3. *The algorithm for $SO(3)$.* Assume $f$ is a band-limited function of band-limit $b$ on $SO(3)$. Then the expression (4.7) is in a perfect form for using the separation of variables idea. The method we use to compute $\hat{f}(V_l)_{m_1,m_2}$ is the obvious one: we sum on $j_1$ first, and then on $j_2$ and $k$ in turn. In other words we compute the following quantities, for $0 \leq j_2, k \leq 2b$ and $|m_1|, |m_2| \leq l \leq b$.

$$F_1(j_2, k; m_1) = \frac{1}{2b+1} \sum_{j_1=0}^{2b} e^{im_1\varphi^{j_1}_1} f(\varphi^{j_2}_2, \theta^k, \varphi^{j_1}_1) \tag{4.8}$$

$$F_2(k; m_1, m_2) = \frac{1}{2b+1} \sum_{j_2=0}^{2b} e^{im_2\varphi^{j_2}_2} F_1(j_2, k; m_1) \tag{4.9}$$

$$\hat{f}(V_l)_{m_1,m_2} = \sum_{k=0}^{2b} w_b(k) P^l_{m_1,m_2}(\theta^k) F_2(k; m_1, m_2) \tag{4.10}$$

If we calculate the sums (4.8) (4.9) (4.10) in a naive manner then the whole calculation takes $(5b+3)(2b+1)^3$ scalar operations. This is the complexity we get from a plain application of the separation of variables technique, and corresponds exactly to the result of Theorem 3.28.

To obtain a more efficient algorithm, we must apply fast transform techniques to the sums (4.8), (4.9), and (4.10). The first two have the form of Fourier transforms on $\mathbf{Z}/(2b+1)\mathbf{Z}$, and so may be efficiently computed by variants of the Cooley-Tukey method. Doing this for all of the sums involved would then require at most $O(b^3 \log b)$ operations. That leaves the final sum (4.10). For this, we must use the fact that for fixed $m_1, m_2$, the functions $P^l_{m_1,m_2}(\cos\theta)$ satisfy a three-term recurrence relation on the index $l$, thereby putting us in a situation in which the fast polynomial transform techniques of Driscoll and Healy [33] and their subsequent generalization [34] apply. These algorithms show that for fixed $m_1, m_2$, we may compute the sums (4.10) for all $0 \leq l \leq 2b$ in $O(b(\log b)^2)$ scalar operations. Thus the total number of operations required to compute all the necessary sums of the form (4.10) is $O(b^3(\log b)^2)$. Putting this together proves the next theorem.

**Theorem 4.6.** *The Fourier transform of a band-limited function, $f$, on $SO(3)$ with band-limit $b$ may be computed in $O(b^3(\log b)^2)$ scalar operations, given the function values of $f$ on the set $X_b$.*

4.2. **The Fourier transform on a compact group.** Many of the arguments and techniques applied in the case of the rotation group and the circle can be

generalized to give an algorithm to compute the Fourier transform of a continuous function defined on any compact group; the definition is a simple extension of that for finite groups. We need to treat the Fourier transforms of both continuous functions and functions sampled on finite subsets of a compact group in a uniform manner. The correct way to do this is via the Fourier transform of a measure on the group.

For the remainder of Section 4 it is convenient for us to fix, for once and for all, a complete set of inequivalent irreducible matrix representations of $G$, which we shall call $\mathcal{R}$. The Fourier transform will be defined as a map from measures on $G$ to vectors in the product space $\mathcal{F}(\mathcal{R}) = \prod_{\rho \in \mathcal{R}} M_{d_\rho}(\mathbf{C})$.

**Definition 4.7** (Fourier transform on a compact group)**.** Assume $G$ is a compact group, $\mu$ is a regular bounded complex Borel measure on $G$.

1. If $\rho$ is any (continuous) finite dimensional matrix representation of $G$, then the Fourier transform of $\mu$ at $\rho$ is
$$\hat{\mu}(\rho) = \int_G \rho(x) d\mu(x) \tag{4.11}$$
2. The **Fourier transform of** $\mu$ is the vector in $\mathcal{F}(\mathcal{R})$ given by
$$\hat{\mu} = \prod_{\rho \in \mathcal{R}} \hat{\mu}(\rho). \tag{4.12}$$
*I.e.,* the element of $\mathcal{F}(\mathcal{R})$ whose $\rho$ coordinate is $\hat{\mu}(\rho)$.
3. Let $\mathcal{R}'$ be any subset of $\mathcal{R}$. Then the Fourier transform of $\mu$ on $\mathcal{R}'$ is the vector in $\mathcal{F}(\mathcal{R})$ whose $\rho$ coordinate is $\hat{f}(\rho)$ when $\rho$ is in $\mathcal{R}'$ and zero otherwise.

Two specific instances of this definition will be of particular concern to us. The first is the Fourier transform of a continuous function $f$. In this case we first multiply $f$ by the Haar measure of unit mass on $G$, which we call $\mu_G$, and then take the Fourier transform of the product $f \cdot \mu_G$. The Fourier transform of $f$ at the representation $\rho$ is then $\hat{f}(\rho) = \int_G f(x)\rho(x)d\mu(x)$, and the Fourier transform on $\mathcal{R}$ or a subset of $\mathcal{R}$ is defined accordingly.

The second case of interest occurs when $\mu$ is a finitely supported measure. In this case, there is a finite subset $X$, of $G$, and a complex valued function $w$, on $X$, such that $\mu = \sum_{x \in X} w(x)\delta_x$, where $\delta_x$ is the Dirac delta distribution at $x$. In this case the Fourier transform of $\mu$ at $\rho$ is a finite sum,
$$\hat{\mu}(\rho) = \sum_{x \in X} w(x)\rho(x). \tag{4.13}$$

4.3. **Sampling.** As in Section 4.1.2 we now need to relate the Fourier transform of a measure constructed from $f$ to the Fourier coefficients of $f$. If we wish to compute the Fourier transform of $f$, then we must usually compute an infinite number of integrals on the group $G$. In practice, even on the fastest computers, we can only compute a finite number of integrals in a finite time, and for numerical computation to be possible, we can only compute finite sums approximating these integrals. In other words, the only information we can hope to compute is the Fourier transform of a finite measure related to $f$ only a finite set of irreducible representations of $G$. The problem of relating the finite information we can compute to the Fourier transform we want to know, is the subject of sampling theory.

There is a class of continuous functions whose Fourier transforms can be computed given only a finite number of function values. These are the functions on $G$ whose expansion in matrix coefficients has only a finite number of nonzero terms. Such functions are called band-limited functions on $G$. For non-band-limited functions on $G$, we may in general only approximate the Fourier transform, and the accuracy of the approximation depends both on the number of function values given and how close the function is to being band-limited. To make this precise, we need to refine the notion of band-limited and define for each positive integer, $b$, a space of "functions with band-limit $b$".

**Definition 4.8** (Band-limits for compact groups)**.** A **system of band-limits on compact group** $G$, is a choice for each non-negative integer $b$, of a finite set of irreducible representations, $\mathcal{R}_b$, contained in $\mathcal{R}$, such that

1. $\bigcup_{b\geq 0} \mathcal{R}_b = \mathcal{R}$
2. If $b_1 \leq b_2$, then $\mathcal{R}_{b_1} \subseteq \mathcal{R}_{b_2}$.
3. If $b_1$ and $b_2$ are any non-negative integers, then $\mathcal{R}_{b_1} \otimes \mathcal{R}_{b_2} \subseteq \mathrm{span}_{\mathbf{Z}}\, \mathcal{R}_{b_1+b_2}$. *I.e.*, the tensor product of a representation in $\mathcal{R}_{b_1}$ with a representation in $\mathcal{R}_{b_2}$ is a sum of representations in $\mathcal{R}_{b_1+b_2}$.

We say that a function on $G$ is **band-limited with band-limit** $b$, if it is a linear combination of matrix coefficients of representations in $\mathcal{R}_b$. A function on $G$ is **band-limited** if it is band-limited with band-limit $b$ for some nonzero integer $b$.

*Remark 4.9.* The definition of band-limited in Definition 4.8 is a special case of a much more general notion. The concept of band-limit may be formulated in the context of any filtered algebra. Even more generally, we may speak of band-limited elements of a filtered module over a filtered algebras (cf. [66]). This is useful for the development of a sampling theory for tensor fields, or sections of vector bundles.

**Example 4.10.** The irreducible representations of $S^1$ are indexed by integers, $j$. If we let $\mathcal{R}_b$ be the set of irreducible representations with $|j| \leq b$, then the collection of sets $\mathcal{R}_b$ is a system of band-limits on $S^1$.

On the group $SO(3)$, we let $\mathcal{R}_b = \{V_l : l \leq b\}$. Then the collection of sets $\mathcal{R}_b$ is a system of band-limits on $SO(3)$.

More generally, if we pick a set of irreducible representations of a compact Lie group which generates the representation ring and contains the trivial representation, we may let $\mathcal{R}_b$ be the set of representations occuring in tensor products of $b$ representations from this set. Then the sets $\mathcal{R}_b$ form a system of band-limits on the given compact Lie group.

The sampling lemma for band-limited functions may now be stated as follows.

**Lemma 4.11** (Band-limited sampling)**.** *Assume $f$ is a band-limited function of band-limit $b$ and $\mu$ is a regular bounded complex measure on $G$ whose Fourier transform at any nontrivial representation in $\mathcal{R}_{2b}$ is 0 and whose Fourier transform at the trivial representation is 1. Then for any representation $\rho$ in $\mathcal{R}_b$,*

$$\widehat{f \cdot \mu}(\rho) = \hat{f}(\rho).$$

*Proof.* Consider the product of $f$ with any term in the Fourier expansion of $\mu$ corresponding to a representation not in $\mathcal{R}_{2b}$. This product must be in the orthogonal complement of the space of functions with band-limit $b$. So, if $\rho$ is in $\mathcal{R}_b$, then $\widehat{f \cdot \mu}(\rho) = \widehat{f \cdot 1}(\rho)$. □

The relevant case of this lemma is when $\mu = \sum_{x \in X} w(x)\delta_x$ is a finitely supported measure. In this case the lemma says that

$$\hat{f}(\rho) = \sum_{x \in X} f(x)w(x)\rho(x)$$

for any $\rho$ in $\mathcal{R}_b$. The measure $\mu$ is then called a **sampling distribution**, and the function $w(x)$ is the associated weight function.

It is always possible to find finitely supported measure that satisfy the conditions of Lemma 4.11. For this to be useful, however, the number of points in the support of this measure must not be too large in comparison with the dimension of the space of functions of band-limit $b$. Another restriction on the set $X$ is that we want to be able to construct a fast transform for functions sampled on this set. In Section 4.4.1 we show that, for the classical groups, it is possible to find sets of points that satisfy all these restrictions at once.

4.3.1. *Functions that are not band-limited.* When the function $f$ appearing in Lemma 4.11 is not band-limited, we may still ask if the Fourier transform of $f \cdot \mu$ is a reasonable approximation to the Fourier transform of $f$. We now give a brief discussion of this question. The ideas introduced here are not required elsewhere in the paper.

If $\mu$ is a measure on $G$, then we may construct a function of band-limit $b$ from $\mu$ by truncating the Fourier expansion of $\mu$ to include only those representations from $\mathcal{R}_b$. Let us call the projection operator corresponding to this process $P_b$. Let $\|\cdot\|_{A_0}$ denote the norm of absolute summability on the space of absolutely summable functions on $G$ (see [52] Section 34.4). Then $\|P_b(f \cdot \mu - f \cdot \mu_G)\|_{A_0}$ measures how close the Fourier transform of $\mu \cdot f$ is to that of $f$, for those representations in $\mathcal{R}_b$. To bound this quantity properly, we now need to be a little more selective in the choice of the sets $\mathcal{R}_b$.

Assume that $G$ is a connected compact Lie group. Then the set of irreducible representations may be identified with the set of all dominant analytically integral weights in the dual of its Cartan subalgebra. Now choose a norm $\|\cdot\|$, defined on the dual of the Cartan subalgebra, such that $\|\cdot\|$ is invariant under the taking of duals, and $\|\alpha\| \leq \|\beta\| + \|\gamma\|$ whenever $\alpha, \beta, \gamma$ correspond to irreducible representations of $G$ such that $\alpha$ is a direct summand of $\beta \otimes \gamma$. Let $\mathcal{R}_b$ be the set of those irreducible representations with norm less than or equal to $b$. Then the collection of sets $\mathcal{R}_b$ is a system of band-limits on $G$. This system of band-limits is very similar to those constructed in Example 4.10.

For any system of band-limits constructed from a norm on the Cartan subalgebra, as above, we have the following theorem. To state it properly, we define the norm, $\|\cdot\|_{A_m}$, which is finite on a subspace of the continuous functions, by

$$\|f\|_{A_m} = \left|\hat{f}(1)\right| + \sum_{\rho \in \mathcal{R} \setminus \{1\}} \|\rho\|^m \|\hat{f}(\rho)\|_{\rho,1}$$

where $\|\ \|_{\rho,1}$ is the von Neumann norm on the space of $d_\rho \times d_\rho$ matrices. (The book [52] develops the general properties of these types of norms.)

**Theorem 4.12** ([66] Theorem 2.11)**.** *Assume $f$ is a continuous function on $G$, and $\mu$ is a regular bounded complex measure whose Fourier transform at any nontrivial representation in $\mathcal{R}_{a+b}$ is 0 and whose Fourier transform at the trivial representation is 1. Let $m$ be and nonnegative integer. Then*

$$\|P_b(f \cdot \mu - f \cdot \mu_G)\|_{A_0} \leq K_G b^{\dim G + \operatorname{rank}_{ss} G} a^{-m} \|\mu\|_1 \|f - P_a f\|_{A_m}$$

*where* $\|\cdot\|_1$ *is the total variation norm, and* $\operatorname{rank}_{ss} G$ *denotes the semisimple rank of* $G$.

Theorem 4.12 may be improved in several different ways. The norms used on the spaces of measures and functions may be generalized. The constants $K_G$ can be determined for specific groups, and the factor $K_G b^{\dim G + \operatorname{rank}_{ss} G} a^{-m}$ appearing in the bound may be improved. The result may also be formulated for sampling using distributions rather than measures, and a version of this result also holds for sections of vector bundles ([66] Theorem 3.7). The work of Gaudry and Pini [40, 76] may be used to relate $\|f - P_a f\|_{A_m}$ to norms based on the differentiability properties of functions.

4.4. **The classical compact Lie groups.** We now consider four series of connected compact Lie groups. The special orthogonal groups, $SO(n)$, the unitary groups, $U(n)$, the special unitary groups, $SU(n)$, and finally the symplectic groups $Sp(n)$. Each of these may be realized as a group of $n \times n$ matrices. In particular, we define $Sp(n)$ to be the set of $n \times n$ quaternionic matrices that are unitary with respect to a quaternionic inner product, i.e. $A^* A = I$. We shall consider these groups of matrices to be embedded in each other in the obvious ways. For example $SO(n) \subseteq U(n) \subseteq U(n+1)$.

Many definitions and simple lemmas that we stated for finite groups apply to compact groups with very little or no change. Rather than restate them in their entirety, let us simply note the minor changes required to accomodate the more general situation.

- The definition of subgroup-adapted sets of representations (Definition 3.7) makes sense for finite sets of matrix representations of a compact group, $G$, and a closed subgroup $H$.
- Lemma 3.9 (Schur's Lemma) holds for $G$ a compact group, and $K, H$ closed subgroups of $G$.
- The notion of maximum multiplicity $\mathcal{M}(G, H)$ makes sense for a compact group $G$ and a closed subgroup $H$.
- The separation of variables equations (3.3) and (3.4) apply to compact groups, provided the function, $f$, is replaced by a finitely supported measure, the elements $g$, which we are factoring, range over the finite support of this measure, and the representations we use are all finite-dimensional. In this way we can formulate the separation of variables technique on any group, compact or not.

With these definitions and results we can describe fast transforms on the classical compact groups in a way very similar to our discussion of $SO(3)$. As in the discussion of $SO(3)$, we start by factoring group elements.

4.4.1. *Generalized Euler angles.* The Euler angle decomposition of an element in the rotation group has a natural generalization to the classical compact Lie groups. For any positive integer, $n$, and real number $\theta$, we define matrices $r_n(\theta)$, $t_n(\theta)$, $q_n(\theta)$, and $u_n(\theta)$ which generate these groups. Let

$$r_n(\theta) = \begin{pmatrix} I_{n-2} & & \\ & \begin{matrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{matrix} & \\ & & I \end{pmatrix}$$

where the rotation block appears in columns $n-1, n$. Let $t_n(\theta)$ be the diagonal matrix with ones on the diagonal except for $e^{i\theta}$ in the $n^{\text{th}}$ diagonal position. Let $q_n(\theta) = t_1(-\theta)\dots t_n(-\theta)t_{n+1}(n\theta)$. Finally, let $u_n(\theta)$ be the diagonal matrix with ones on the diagonal except for $e^{j\theta}$ in the $n^{\text{th}}$ diagonal position, where $j$ is a unit quaternion.

**Lemma 4.13** (Generalized Euler angles)**.** *Let $g_{SO(n)}$, $g_{U(n)}$, $g_{SU(n)}$, and $g_{Sp(n)}$ be elements of the groups $SO(n)$, $U(n)$, $SU(n)$, and $Sp(n)$ respectively. Then these group elements have factorizations of the form*

$$\begin{aligned}
g_{SO(n)} &= \left[r_2(\theta_2^n)\cdots r_n(\theta_n^n)\right]\cdots\left[r_2(\theta_2^3)r_3(\theta_3^3)\right]\cdot r_2(\theta_2^2)\\
g_{U(n)} &= \left[t_1(\varphi_1^n)\cdots t_{n-1}(\varphi_{n-1}^n)r_2(\theta_2^n)\cdots r_n(\theta_n^n)\cdot t_n(\varphi_n^n)\right]\cdot\\
&\qquad\qquad\cdots\left[t_1(\varphi_1^2)r_2(\theta_2^2)t_2(\varphi_2^2)\right]\cdot t_1(\varphi_1^1)\\
g_{SU(n)} &= \left[q_1(\varphi_1^n)\cdots q_{n-1}(\varphi_{n-1}^n)r_2(\theta_2^n)\cdots r_n(\theta_n^n)\cdot q_{n-1}(\varphi_n^n)\right]\cdot\\
&\qquad\qquad\cdots\left[q_1(\varphi_1^2)r_2(\theta_2^2)q_1(\varphi_2^2)\right]\\
g_{Sp(n)} &= \left[\left(t_1(\varphi_{1,1}^n)u_1(\psi_1^n)t_1(\varphi_{2,1}^n)r_2(\theta_2^n)\right)\cdots\right.\\
&\quad\left.\cdots\left(t_{n-1}(\varphi_{1,n-1}^n)u_{n-1}(\psi_{n-1}^n)t_{n-1}(\varphi_{2,n-1}^n)r_n(\theta_n^n)\right)\cdot t_n(\varphi_{1,n}^n)u_n(\psi_n^n)\cdot t_n(\varphi_{2,n}^n)\right]\cdot\\
&\qquad\qquad\cdots\left[t_1(\varphi_{1,1}^1).u_1(\psi_1^1).t_1(\varphi_{2,1}^1)\right]
\end{aligned}$$

For a proof of Lemma 4.13 see [60] or [65]. The sets over which the parameters $\theta_j^k, \varphi_j^k, \psi_j^k, \varphi_{a,j}^k$ should range can easily be determined from the interpretation of the factors in these expressions as double coset representatives. For example, if $k \geq 3$ then $r_k([0,\pi])$ is a set of double coset representatives for $SO(k)$ relative to $SO(k-1)$. Therefore $\theta_k^n$ should range over the set $[0,\pi]$, when $k \geq 3$. Similarly, $r_n([0,\frac{\pi}{2}])$ is a set of double coset representatives for $U(n)$ relative to $U(n-1)\times U(1)$, and $u_n([0,\frac{\pi}{2}])$ is a set of double coset representatives for $Sp(n)$ relative to $Sp(n-1)\times Sp(1)$.

As in the case of the rotation group, we use the generalized Euler angle factorization to specify the sets we shall use for sampling. In each case we pick a set $X_b$ which is a grid in the coordinate systems determined by the factorizations of Lemma 4.13, with only $O(b)$ points in each dimension. To find suitable weight functions $w_b$, and hence suitable measures $\mu_b$ for the band-limited sampling lemma to apply, it suffices to find quadrature rules which exactly integrate spherical function on the double coset spaces related to the factorizations. This method is most easily stated using the convolution of measures, as in the next lemma.

**Lemma 4.14.** *Assume $G$ is a compact group $K$ is a closed subgroup of $G$, and $\mathcal{R}'$ is a set of irreducible representations of $G$. Assume that $\chi$ is a measure supported on $K$, such that the Fourier transform of $\chi - \mu_K$ on $\mathcal{R}'$ is zero, and $\nu$ is a measure on $G$ such that the integral of $\nu - \mu_G$ against any zonal spherical function on $G$ relative to $K$ for a representation in $\mathcal{R}'$ is zero. Let $\mu = \chi * \nu * \chi$. Then the Fourier transform of $\mu$ at any nontrivial representation in $\mathcal{R}'$ is zero, and the Fourier transform of $\mu$ at the trivial representation is* 1*.*

4.4.2. *Fast transforms.* Two things remain to be done before we can state a result about the efficient computation of Fourier transforms on the classical compact Lie groups. We must define a system of band-limits on each of the classical groups, and we must specify subgroup chains for our representations to be adapted to.

A specific choice of a system of band-limits can be obtained by writing the highest weights of representations in a standard coordinate system. Representations of the

special orthogonal, unitary, special unitary or symplectic groups may be indexed by $r$-tuples of integers satisfying certain inequalities depending on the group, where $r$ is the rank (see e.g. [60]). For example, the representations of $SO(2r)$ are indexed by integers $m_1, \ldots, m_r$, which satisfy

$$m_1 \geq \cdots \geq m_{r-1} \geq |m_r| .$$

For any of the series of groups under consideration, we define a system of band-limits by letting $\mathcal{R}_b$ be the set of those representations for which all integers, $m_i$ in the corresponding $r$-tuple satisfy $|m_i| \leq b$. When the group is $SO(2r)$, this means simply that $\mathcal{R}_b$ is the set of all representations with $m_1 \leq b$.

Finally, we specify the subgroup chains which our representations must be adapted to. These are:

$$SO(n) > SO(n-1) > \cdots > SO(2), \tag{4.14}$$

$$U(n) > U_{n-1} \times U_1 > \cdots > U(1), \tag{4.15}$$

$$SU(n) > S(U_{n-1} \times U_1) > \cdots > S(U_1 \times U_1) \tag{4.16}$$

$$Sp(n) > Sp(n-1) \times Sp(1) > Sp(n-1) \times U(1) > \cdots > Sp(1) > U(1). \tag{4.17}$$

Now we have all pieces needed to apply the separation of variables techniques to these groups. We may apply both the matrix and scalar methods, but, as in the case of finite groups, the scalar method produces better results. The next theorem uses the scalar method.

**Theorem 4.15.** *[[64] Theorem 5.1] Let $K_n$ be one of the groups, $SO(n)$, $U(n)$, $SU(n)$, or $Sp(n)$. Then the Fourier transform of a band-limited function $f$ on $K_n$ with band-limit $b$ may be computed at a set of representations adapted to the subgroup chain (4.14)-(4.17) for $K_n$, in $O(b^{\dim K_n + \gamma(K_n)})$ scalar operations, given the function values of $f$ on the set $X_b$ . The exponent $\gamma(K_n)$ is given in the following table.*

| $K_n$ | $SO(n)$ | $U(n)$ | $SU(n)$ | $Sp(n)$ |
|---|---|---|---|---|
| $\gamma(K_n)$ | $\lfloor \frac{n}{2} \rfloor$ | $n-1$ | $n$ | $3n$ |

The proof of Theorem 4.15 follows the $SO(3)$ case very closely. Using Schur's Lemma and the factorization of elements in $X_b$ into Euler angles, we obtain an expression for the Fourier transform, in coordinates, as an iterated sum of products. We then calculate the sums corresponding to different factors one after the other. To determine which matrix elements occur multiplied by each other in these expressions, we index the rows and columns of our representations by paths in a Bratteli diagram, and use the diagrammatic methods of Sections 3.3.1 and 3.3.2. For a proof, see [64].

The matrix elements of $SO(n)$ in a Gel'fand-Tsetlin basis were first determined in [61]. A general method of finding expressions for matrix elements is presented, and explicit expressions for the unitary groups may be found in [60].

4.4.3. *Relationship to the finite case.* The fast Fourier transform on $SO(n)$ has an interesting relation to some finite Fourier transforms. The fast transform on this group depends on two properties: The form of factorization into generalized Euler angles, and the structure of the representation matrices of the elements $r_n(\theta)$. Both of these may be deduced from the following definition.

**Definition 4.16** (Two-step commuting chain)**.** A **two-step commuting chain** for a compact group $G$, is a chain of closed subgroups,

$$G = K_n \geq K_{n-1} \geq \cdots \geq K_0 = 1 \tag{4.18}$$

such that there exist subsets $A_1, \ldots, A_n$ with the following properties:

(1) $A_i \subseteq K_i$.
(2) $K_i = K_{i-1} \cdot A_i \cdot K_{i-1}$.
(3) The elements of $A_i$ centralize $K_{i-2}$.

**Lemma 4.17.** *Assume $G$ has a two-step commuting chain. Then*

$$G = (A_1 \cdots A_n) \cdot (A_1 \cdots A_{n-1}) \cdots A_1$$

Lemma 4.17 says that any element of $G$ may be factored using elements from the sets $A_i$. On the other hand, following Section 3.3, properties (4.16) and (4.16) in the definition of a two-step commuting chain indicate that the nonzero entries of adapted representation matrices at elements coming from the sets $A_i$ may be indexed by pairs of paths of length 2 in the Bratteli diagram of the chain (4.18). Thus, the scalar and matrix separation of variables algorithm on $SO(n)$ may be formulated in any group with a two-step commuting chain.

There are many groups with two-step commuting chains. The classical compact groups, $SO(n)$, $U(n)$, $SU(n)$, and $Sp(n)$, all have obvious two-step commuting chains. It is particularly interesting that there are finite groups which have two-step commuting chains. Indeed, the chains of finite groups

$$S_n > S_{n-1} > \cdots > S_1 = 1$$
$$GL_n > GL_{n-1} > \ldots GL_1 > 1$$

are both examples of two-step commuting chains which, as Section 3.1 shows, lead to fast Fourier transform algorithms.

4.4.4. *Multimatrix algebras and separation of variables.* The ideas used in separation of variables algorithms may be formulated for an arbitrary multi-matrix algebra, *i.e.,* a direct product of matrix algebras. We may still factor elements, and the structure of matrices may still be described using paths in Bratteli diagrams, so all the basic ingredients for this algorithm are there.

The results we have developed for finite groups and compact groups are both special cases of a separation of variables algorithm on a finite multi-matrix algebra. In the finite group case, the group algebra is semi-simple and hence isomorphic to a multi-matrix algebra. In the compact group case, we must work with a finite dimensional quotient of the group algebra by an ideal. The sampling theorems then tell us how to relate these computations to Fourier transforms of functions. From this point of view, the compact group and finite group algorithms are part of the same phenomenon.

In the compact group case, we chose only one of many possible ways of completing a direct sum of matrix algebras. By considering different completions we may describe fast methods for computing sums of products in algebras of distributions; once we have used sampling theory to divide by an ideal the algorithms are the same as before.

So far we have considered sampling to be equivalent to multiplying a function by a measure that approximates Haar measure. We may also sample a function by multiplying by a finitely supported distribution. By computing with this product,

we may compute Fourier transforms of functions given values of their derivatives, as well as function values. We may also consider distributions that are left invariant under the action of some subgroup. This is equivalent to computing the Fourier transform of distributions on a homogeneous space of the group.

4.5. **Homogeneous spaces.** As in the finite case (cf. Section 3.1.5) the homogeneous spaces of compact Lie groups provide examples particularly amenable to the separation of variables technique. As a function on a homogeneous space $G/K$ may be viewed as a right-$K$-invariant function on the group $G$, all the sampling theory developed for groups is readily transferred to the new setting. The idea of factoring group elements is translated to factorization of coset representatives for $G/K$, but as many of our choices of factorization came from coset representatives in the first place, this is not usually a big change.

Not all of the Fourier coefficients of a right-$K$-invariant function are relevant. In fact only certain Fourier coefficients may be nonzero. The matrix coefficients corresponding to the possible nonzero Fourier coefficients are precisely the matrix coefficients that are themselves right-$K$-invariant, and hence may be interpreted as functions on $G/K$. These functions are called the associated spherical functions on $G/K$, and the Fourier transform of a function on $G/K$.

4.5.1. *Rank one homogeneous spaces.* In one important class of examples, the classical rank one homogeneous spaces, the associated spherical functions factor completely into products of orthogonal polynomials and complex exponentials.

For example, the associated spherical functions for $S^{n-1} = SO(n)/SO(n-1)$ may be written, in spherical polar coordinates, and in an appropriate basis, as

$$A^n_{\mathbf{M}} \left[ \prod_{j=3}^{n} C^{\frac{j-2}{2}+|m_{j-1}|}_{m_j-|m_{j-1}|}(\cos\theta_j) \sin^{m_{j-1}}(\theta_j) \right] \cdot e^{im_2\theta_2}$$

where $A^n_{\mathbf{M}}$ is a constant depending on $n$ and the integers $m_i$. In these cases, the technique of summing one coordinate at a time may be supplemented by the algorithms of Driscoll and Healy to produce an even more efficient algorithm for computing the transform. It follows that we may expand a band-limited function of band-limit $b$ on $S^{n-1}$ in a basis of associated spherical functions in $O(b^{n-1}(\log b)^2)$ scalar operations.

The algorithms of Driscoll and Healy were originally developed to treat such a case: the spherical harmonics are precisely the associated spherical functions for $S^2 = SO(3)/SO(2)$. Further examples of this method applied to rank one homogeneous spaces may be found in [65, 64].

4.5.2. *The expansion of tensor fields.* The definition of Fourier transform may be extended to treat the expansion of sections of a homogenous vector bundle in terms of basis sections coming from the decomposition of sections of that bundle under the action of a compact group. As with the case of homogeneous spaces, the sampling theory for compact groups may be adapted to this situation, and it therefore makes sense to ask for efficient algorithms for computing the expansion of a band-limited section in basis sections.

An example of this would be the expansion of tensor fields on $S^2$ in terms of tensor spherical harmonics. The tensor harmonics that are sections of irreducible vector bundles over $S^2$ are called the monopole tensor harmonics. The monopole

tensor harmonics on the irreducible bundle of degree $n$ may be written in appropriate bases away from the poles, as linear combinations of scalar spherical harmonics times $1/(\sin\theta)^{|n|}$. This observation quickly reduces the problem of finding a fast tensor harmonic transform to the well-known scalar case treated by Driscoll and Healy. For a more detailed treatment of the tensor transform, see [65, 48].

## 5. Fast discrete polynomial transforms

The techniques we have described in Sections 3 and 4 reduce the computation of Fourier transforms on groups to sums involving matrix coefficients, taken over fixed subsets of the group. Many well-known special functions arise in this way, as matrix coefficients considered on a subset of a group, so it is sometimes possible to use the special function properties, recurrences etc., of the matrix coefficients to help compute the new sums efficiently. For example, we have seen that for the $n$-sphere, the Gegenbauer polynomials arise. In this case a pure separation of variables approach to computing the transform of a band-limited function takes $O(b^{n+1})$ operations, whereas an approach that uses fast Gegenbauer transforms takes $O(b^n(\log b)^2)$ operations. Here we summarize some of the recent work [33, 34, 73] on fast discrete polynomial transforms, especially as related to these new fast algorithms for discrete orthogonal polynomial transforms.

The general framework is as follows. Let $\{P_0, \ldots, P_{N-1}\}$ denote a set of linearly independent polynomials with complex coefficients. Let $\{z_0, \ldots, z_{N-1}\} \subset \mathbf{C}$ be any set of $N$ distinct points, called sample points. If $f = (f_0, \ldots, f_{N-1})$ is any data vector, then the **discrete polynomial transform** of $f$, the set is defined as the collection of sums, $\{\hat{f}(P_0), \ldots, \hat{f}(P_{N-1})\}$, where

$$\hat{f}(P_j) = \sum_{i=0}^{N-1} f_i P_j(z_i) w(i). \tag{5.1}$$

The function $w$ is some associated weight function, which we will usually take to be identically 1. It is easy to see that a general discrete polynomial transform may be written as a matrix-vector product, so direct computation requires on the order of $N^2$ operations.

*Remark 5.1.* There are at least four distinct transforms that may be associated with a linearly independent sequence of polynomials and a set of points.

(1) Given the coefficients of a polynomial $f$ in the basis $\{P_k\}$, evaluate $f$ at the points $\{z_j\}$.
(2) Given a sequence of function values $f_j = f(z_j)$, of a polynomial $f$, compute the coefficients of the expansion of $f$ in the basis $\{P_k\}$. This transform is inverse to the evaluation transform (1).
(3) The transpose of (1). This transform is equivalent to (5.1) in the case where the weight function $w$ is identically 1.
(4) The transpose of (2).

The discrete polynomial transform may be related to either transform (3) or in certain cases to (2). On the other hand, it is clear that by multiplying a data vector by the weight function we can reduce a discrete polynomial transform to the transposed evaluation (3).

**Example 5.2** (Discrete monomial transforms)**.** When the $P_k(z) = z^k$ is the monomial of degree $k$, we obtain a discrete monomial transform for which an $O(N \log^2 N)$

algorithm exists (see Section 5.2). If the sample points all lie on the unit circle, then with an appropriate weight function the discrete monomial transform is a Fourier transform on the circle. If, in addition, the sample points are the roots of unity then we obtain the discrete Fourier transform.

**Example 5.3** (Discrete cosine transform)**.** The Chebyshev polynomials of the first kind are the sequence of orthogonal polynomials defined recursively by $T_k(z) = 2zT_{k-1}(z) - T_{k-2}(z)$, with the initial conditions $T_0 = 1$, $T_1 = z$. The discrete cosine transform (DCT) is the discrete polynomial transform for the Chebyshev polynomials, with weight function 1, and sample points $z_j = \cos^{-1}(\pi(2j+1)/2N)$. This transform may be computed in $O(N \log N)$ operations using an FFT [1], or by the extremely efficient DCT algorithm of Steidl and Tasche [86], which takes a mere $(3/2)N \log N - N + 1$ real scalar operations when $N$ is a power of 2 and the input vector is real. The sample points we have used here are called the **Chebyshev points** and are the roots of $T_N$.

**Example 5.4** (Discrete orthogonal polynomial transforms)**.** We have already seen that the Fourier transforms on the $k$-sphere may be computed using discrete polynomial transforms at the Gegenbauer polynomials. In this case, the set of polynomials is orthogonal with respect to a measure on the unit interval, and as is discussed in the next section, $O(N \log^2 N)$ algorithms exist [33, 34].

**Example 5.5** (Discrete spherical transforms)**.** The spherical functions of distance transitive graphs may be written in terms of orthogonal polynomials (see [85, 35]). The paper [34] develops an algorithm for computing the spherical transform on these spaces.

5.1. **Fast discrete orthogonal polynomial transforms.** In [34] an algorithm is given which computes general **discrete orthogonal polynomial transforms** in $O(N(\log N)^2)$ operations. By this we mean a discrete polynomial transform in which the $P_k$ are a set of orthogonal polynomials with $\deg P_k = k$. The algorithm relies on the three-term recurrences satisfied by any orthogonal polynomial system, and therefore may be used for computing transforms over any set of spanning functions which satisfy such a recurrence. Related techniques have already found a number of applications attacking computational bottlenecks in problems in areas such as medical imaging, geophysics and matched filter design [34, 72, 49, 48]. This general approach grew out of the fast Legendre transform algorithms in [33] (esp. Section 5).

The main result of [34] is the following theorem.

**Theorem 5.6.** *Let $N = 2^r$ and let $P_l(x)$, $l = 0, \ldots, N-1$ comprise a family of functions defined at the positive integers $x = 0, 1, \ldots, N-1$ and satisfying a three-term recurrence there:*

$$P_{l+1}(x) = (a_l x + b_l)P_l(x) + c_l P_{l-1}(x), \tag{5.2}$$

*with initial conditions $P_0 = 1$, $P_{-1} = 0$. Then the discrete polynomial transforms $\{\hat{f}(0), \ldots, \hat{f}(N-1)\}$ for $f = (f_0, \ldots, f_{n-1})$ defined by*

$$\hat{f}(l) = \sum_{j=0}^{N-1} f_j P_l(j) w_j$$

*where $w$ is a weight function, can be computed in $O(N(\log N)^2)$ operations.*

Rather than prove this general theorem, let us sketch the proof of a special case.

**Lemma 5.7.** *Assume $N = 2^r$ and $\{P_0, \ldots P_{N-1}\}$ is a set of real orthogonal polynomials. Then for any sequence of real numbers, $f_0, \ldots, f_{N-1}$, the discrete orthogonal polynomial transform defined by the sums*

$$\hat{f}(l) = \frac{1}{N} \sum_{j=0}^{N-1} f_j P_l \left( \cos^{-1}(\frac{\pi(2j+1)}{2N}) \right) \tag{5.3}$$

*may be calculated in less than*

$$\frac{9}{4} N (\log_2 N)^2 + \frac{1}{4} N \log N + \frac{9}{2} N - 7$$

*real scalar operations.*

The frst stage in the proof of this lemma is to rewrite the sums (5.3), using discrete cosine transforms. To do this, we define the **truncation operators** $\mathcal{T}_n$ which map polynomials of any degree to polynomials of degree strictly less than $n$. If $f = \sum_{k \geq 0} b_k T_k$ is a polynomial, of any degree, written in the basis of Chebyshev polynomials, then let

$$\mathcal{T}_n f = \sum_{k=0}^{n-1} b_k T_k$$

In other words, $\mathcal{T}_n f$ is obtained from $f$ by throwing away terms of degree $n$ or higher in the expansion of $f$ in Chebyshev polynomials.

Now let $f$ be the polynomial of degree $N-1$ uniquely determined by the requirement that

$$f \left( \cos^{-1}(\frac{\pi(2j+1)}{2N}) \right) = f_j \,, \tag{5.4}$$

*i.e.,* by its function values at a set of Chebyshev points. Then the discrete orthogonal polynomial transform of the sequence $\{f_j\}$, may be written as

$$\hat{f}(l) = \mathcal{T}_1 (f \cdot P_l)$$

*Proof of Lemma 5.7.* We start with the polynomial $f$ of degree $N-1$ determined by equation 5.4. As this polynomial is determined by its function values at a set of Chebyshev points, we agree that any calculations involving this polynomial should start with these function values as input data. Note that with $f$ in this representation, the truncation operators $\mathcal{T}_n$ may be applied in $O(N \log N)$ operations using a fast DCT.

The algorithm works by computing intermediate polynomials $Z_l^K = \mathcal{T}_K (f \cdot P_l)$, of degree at most $l-1$, for various values of $l$ and $K$. Without loss of generality, we assume that $P_0 = 1$, so in this notation $\hat{f}(l) = Z_l^1$. We use two basic facts to derive recurrence relations between the $Z_l^K$. First, by iterating the three-term recurrence relation for the polynomials $P_l$, we obtain the recurrence

$$P_{l+m} = Q_{l,m} . P_l + R_{l,m} P_{l-1} \tag{5.5}$$

where $Q_{l,m}$ and $R_{l,m}$ are polynomials of degrees $m$ and $m-1$ respectively. Secondly, the truncation operators satisfy the following "aliasing" property. If $h$ is any polynomial and $Q$ is a polynomial with degree $\deg Q \leq m \leq K$, then

$$\mathcal{T}_{K-m} (h \cdot Q) = \mathcal{T}_{K-m} [(\mathcal{T}_K f) \cdot Q] \tag{5.6}$$

Using equations (5.5) and (5.6), it is trivial to derive the following recurrences for the $Z_l^K$.

$$Z_{l+m}^{K-m} = \mathcal{T}_{K-m}[Z_l^K \cdot Q_{l,m} + Z_{l-1}^K \cdot R_{l,m}] \tag{5.7}$$

$$Z_{l+m-1}^{K-m} = \mathcal{T}_{K-m-1}[Z_l^K \cdot Q_{l,m} + Z_{l-1}^K \cdot R_{l,m}] \tag{5.8}$$

The algorithm proceeds in $r+1 = \log_2 N + 1$ stages as follows.

*Stage* 0. We start at stage 0 by using the formulas $Z_0^N = \mathcal{T}_N f = f$ and $Z_1^N = \mathcal{T}_N(f \cdot P_1)$ to find the polynomials $Z_0^N$ and $Z_1^N$ at the Chebyshev points for $T_N$.

*Stage* 1. At stage 1, we find the polynomials $Z_0^{N/2}, Z_1^{N/2}$ and $Z_{N/2}^{N/2}, Z_{N/2+1}^{N/2}$. We do this using the relations $Z_0^{N/2} = \mathcal{T}_{N/2} Z_0^N$, $Z_1^{N/2} = \mathcal{T}_{N/2} Z_1^N$, and the following recurrences obtained from (5.7) and (5.8)

$$Z_{N/2+1}^{N/2} = \mathcal{T}_{N/2}[Z_1^N \cdot Q_{1,N/2} + Z_0^N \cdot R_{1,N/2}]$$

$$Z_{N/2}^{N/2} = \mathcal{T}_{N/2}[Z_l^N \cdot Q_{1,N/2-1} + Z_0^N \cdot R_{1,N/2-1}]$$

The multiplications may be performed using function values at the Chebyshev points for $T_N$.

*Stage* $k > 1$. At stage $k > 1$ of the algorithm we find the polynomials $Z_l^{N/2^k}$, $Z_{l-1}^{N/2}$, for $l = p(N/2^k) + 1$, $0 \le p < 2^k$, *i.e.*,

$$Z_0^{N/2^k}, Z_1^{N/2^k} \quad Z_{N/2^k}^{N/2^k}, Z_{N/2^k+1}^{N/2^k}, \quad \dots \quad Z_{N-N/2^k}^{N/2^k}, Z_{N-N/2^k+1}^{N/2^k}.$$

These polynomials are determined by finding their values at the $N/2^k$ Chebyshev points for $T_{N/2^k}$. When $p$ is even we simply use the relations $Z_l^{N/2^k} = \mathcal{T}_{N/2^k} Z_l^{N/2^{k-1}}$ and $Z_{l-1}^{N/2^k} = \mathcal{T}_{N/2^k} Z_{l-1}^{N/2^{k-1}}$.

When $p = 2q+1$ is odd we find $Z_{pN/2^k+1}^{N/2^k}$ from recurrence (5.7) by substituting $K = N/2^{k-1}$, $l = qN/2^{k-1} + 1$, $m = N/2^k$, and we find $Z_{pN/2^k}^{N/2^k}$ from recurrence (5.8) by substituting $K = N/2^{k-1}$, $l = qN/2^{k-1} + 1$, $m = N/2^k - 1$. In each case we may perform the polynomial multiplications by multiplying function values and we may apply the truncation operators using DCTs.

*Stage* $r$. At the final stage, $r$, we compute the polynomials $Z_l^1$ for $O \le l < N$. These polynomials are constant, and $\hat{f}(l) = Z_l^1$.

If we are careful about the data we store at each stage, then at every stage, except stage 0, we use $2^k$ DCTs, $2^k$ inverse DCTs, and an additional $4N$ real scalar operations. Adding up all the operations gives the result. □

Figure 10 illustrates the organization of the computation. We have only been able to give the briefest summary of this algorithm. The interested reader is strongly advised to refer to [33, 34, 49], for a more detailed discussion of the algorithm, and particularly to [72, 73] for a discussion of stability issues. [67] contains an axiomatic derivation.

With the exception of the zeroth and last stages, the algorithm proceeds by divide and conquer; the $k$-th stage looks like 2 copies of the $(k-1)$-th stage. The algorithm may also be formulated as a factorization of the matrix $[P_l(z_j)]$, where

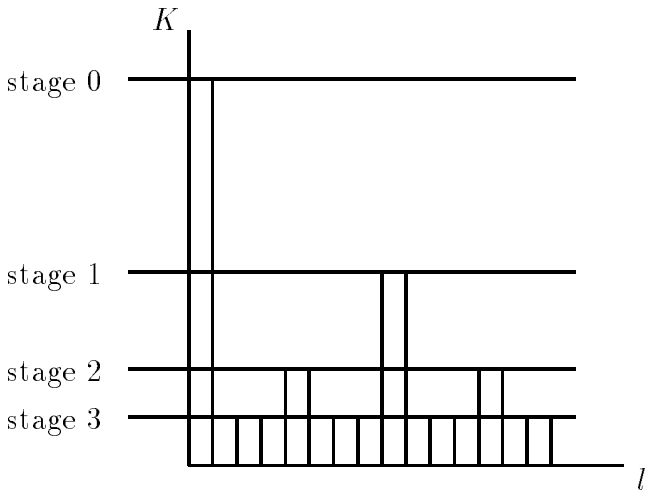



FIGURE 10. Depiction of the computation of $Z_l^K$ for $N = 16$.

$z_j$ are the Chebyshev points, into a product of block matrices with Toeplitz blocks of geometrically decreasing size.

The main step in reducing Theorem 5.6 to Lemma 5.7 is a **fast monomial transform**. We discuss this briefly in the following section.

5.2. **Fast monomial transforms.** The case of monomial transforms, *i.e.*, that in which $P_j(x) = x^j$, is itself quite important and interesting. As noted above, for appropriate evaluation points this formulation produces both the DFT and DCT. Another important set of evaluation points are an arbitrary set of points on the unit circle in which case so-called "non-uniform" DFTs can be obtained. Such transforms have important applications in medical imaging (see eg. [94]).

For sample points $\{z_0, \ldots, z_{n-1}\}$ and initial data $\{f(0), \ldots, f(n-1)\}$, the discrete monomial transform is the computation of the sums

$$\hat{f}(j) = \sum_{k=0}^{n-1} f(k) z_k^j \tag{5.9}$$

or equivalently the computation of the matrix-vector product

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ z_0 & z_1 \cdots & z_{n-1} & \\ z_0^2 & z_1^2 \cdots & z_{n-1}^2 & \\ \vdots & \vdots & \ddots & \vdots \\ z_0^{n-1} & z_1^{n-1} \cdots & z_{n-1}^{n-1} & \end{pmatrix} \cdot \begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(n-1) \end{pmatrix}. \tag{5.10}$$

The fast algorithms of [34, 73] are derived by consideration of the transposed computation, which is simply the evaluation of the polynomial $f(0) + f(1)x + f(2)x^2 + \cdots + f(n-1)x^{n-1}$ at the points $\{z_0, \ldots, z_{n-1}\}$. By formulating an $O(n \log^2 n)$ algorithm of Borodin and Munro [14] as a factorization of the Vandermonde matrix (the transpose of the matrix in (5.10)) an algorithm of equal complexity is obtained for the discrete monomial transform. The paper [73] also

addresses stability issues. In the case of nonuniform DFTs, approximate algorithms using fast multipole expansions have also been explored [36, 37].

## 6. Open questions and further research directions

We hope that the preceding discussions have supported the claim that "Generalized FFTs" is an exciting interdiscplinary research area. Many open questions and potential research directions still remain. We close with a list of some of these, given in no particular order of preference.

**1. Classification of finite groups by complexity.** Is there a universal constant $c$ such that all finite groups have $O(G \log^c G)$ complexity? Is $c = 1$? Within this problem, there are the specific subproblems of finding other families of groups that achieve this upper bound. This would include improving the upper bounds for the various matrix groups over finite fields and Lie groups of finite type, cf. [68, 69, 62]) as well as solvable groups [12].

**2. Improve the general upper bound for finite groups.** Can the general upper bound for the complexity of a finite group be improved? To date, it is known that all groups have complexity bounded $|G|^{1.44}$. This is due to M. Clausen (see [16]). Perhaps the centralizer techniques of [70, 68, 69] will prove useful.

**3. Use of group extensions.** The goal here would be to relate, for $G \triangleright N$, the complexity of $G$, to the complexity of $N$ and $G/N$ for arbitrary $G/N$. As stated in Section 3.5, to date, only the case of $G/N$ abelian has been analyzed [83], as well as those semidirect products that arise as wreath products [81]. Investigating arbitrary semidirect products might be a good place to start.

**4. Generalization of the chirp-$z$ transform.** Most of the FFTs for finite groups use in one way or another, the restriction of representations from group to subgroup. We saw in Section 3.4, that the method of Rader and the chirp-$z$ transform provide ways of relating Fourier analysis on $\mathbf{Z}/p\mathbf{Z}$ to that of $\mathbf{Z}/p\mathbf{Z}^\times \cong \mathbf{Z}/(p-1)\mathbf{Z}$ or a higher order cyclic group $\mathbf{Z}/M\mathbf{Z}$, through convolution. This seems to relate the representation theory of groups that have no group-subgroup relationship. It would be of great interest to find a more general formulation of this, i.e. a chirp-$z$ for nonabelian groups.

**5. Using the Clebsch-Gordan formula.** The Clebsch-Gordan formula appears, somewhat disguised, in the algorithms for solvable groups as well as the Driscoll-Healy algorithm and the fast discrete orthogonal polynomial transforms (where it may be used to derive the three-term recurrence relations). Is it possible to use this formula more completely, possibly achieving an $O(N \log N)$ algorithm?

**6. Monomial representations.** Can Baum's method for supersolvable groups work for other solvable groups? This is already known for extensions of abelian groups with supersolvable factor group [19]. A good place to start would be to look at other $M$-groups which are not supersolvable.

**7. Generating irreducible matrix coefficients.** Any implementation of an FFT would require the ability to actually generate a collection of irreducible matrix coefficients, possibly with some extra structure (e.g., adaptedness). While general results exist in this direction (cf. [5]), it would be of interest to find other classes of

groups which have finely tuned algorithms for this task, as do the solvable groups [8].

**8. Stability.** Error analyses of abelian FFTs abound, mainly due to the widespread use of these algorithms in so many important applications (see e.g. [17, 79]). The audience and usefulness of generalized FFTs is growing and analogous analyses for FFTs for arbitrary or specific classes of groups might be a worthwhile topic to pursue.

**9. Fast algorithms for higher-dimensional recurrences.** The key to the fast discrete polynomial transforms is the use of the three-term recurrence [33, 34]. Can methods like these work for higher-dimensional recurrences? (E.g., the two-dimensional recurrences for spherical functions on rank 2 homogeneous spaces). In the same spirit, it should be noted that the $O(n \log^2 n)$ FFT for the two-sphere [33] uses only the recurrence in the order parameter for the associated Legendre functions. Perhaps in combination with other recurrences that involve the degree parameter, this complexity could be reduced to $O(n \log n)$.

**10. Minimal sampling.** How can we find sampling distributions for band-limited functions of a fixed band-limit with a minimal number of points? This problem seems to have a more algebro-geometric flavour (see [71]).

**11. General sample points.** Can we find fast transforms on compact groups that use more general sampling points? A solution to this problem would be particularly useful on the sphere. There is already some progress in this direction, as the separation of variables technique does not strictly require a grid of points to work, but only the condition that the projection of the set of sample points onto the first few Euler angles (with respect to some fixed ordering of these coordinates) does not contain too many points. However it is not clear whether the sample sets we can treat in this way are general enough to construct minimal or near minimal sample distributions.

**12. Transforms on noncompact groups.** Is it possible to develop algorithms for computing Fourier expansions on noncompact Lie groups or their homogeneous spaces? This would require a sensible sampling theory for these groups. One application of this might be to analyse sets of covariance matrices using harmonic analysis on the space of positive definite matrices. Another example of interest is the motion group, for which there would be many possible applications to filter design and pattern recognition.

**13. Applications.** Generalized FFTs are slowly finding areas of application. The paper [80] attempts to survey some of these. Just to indicate a few directions of work, FFTs for finite groups have found uses in data analysis (see e.g., [28, 29]), finite element methods [39] and filter design [57, 53, 54]. The FFT on the two-sphere has potential applications in global circulation modeling [48], control theory [46, 47] and computer vision [56]. Both investigating these current applications more thoroughly, as well as finding new uses, seem to be worthwhile research directions.

**14. Parallelizability.** The divide and conquer nature of several of the techniques explained here (eg. separation of variables, discrete orthogonal polynomial transforms) strongly suggest the possibility of effective parallel implementation. For finite groups, the recent thesis of L. Stiller [87] takes an excellent first step in this

direction, developing parallel algorithms based on [70]. These algorithms were motivated by their potential use in certain large-scale linear algebra problems. The possible uses of the FFT on the 2-sphere similarly suggest that parallel versions of the compact group FFTs would be of interest. To date only the preliminary work [50] exists.

## References


[1] N. Ahmed, T. Natarajan, and K. Rao, *Discrete cosine transforms*, IEEE Trans. Comput. **23** (1974), 90–93.

[2] B. Alpert and V. Roklin, *A fast algorithm for the evaluation of Legendre transforms*, SIAM J. Sci. Statist. Comput. **12** (1991), 158–179.

[3] L. Auslander and R. Tolmieri, *Is computing with the fast Fourier transform pure or applied mathematics?*, Bull. Amer. Math. Soc. (N.S.) **1** (1979), 847–897.

[4] L. Auslander, J. R. Johnson and R. W. Johnson, *Multidimensional Cooley-Tukey algorithms revisited*, 1995, (preprint).

[5] L. Babai and L. Rónyai, *Computing irreducible representations of finite groups*, Math. Comp. **55** (1990), 705–722.

[6] L. Babai, K. Freidl, and M. Stricker, *Decomposition of *-closed algebras in polynomial time*, Proc. $18^{th}$ ACM ISSAC (1993), 86–94.

[7] U. Baum, *Existence and efficient construction of fast Fourier transforms for supersolvable groups*, Comput. Complexity **1** (1991), 235–256.

[8] U. Baum and M. Clausen, *Computing irreducible representations of supersolvable groups*, Math. Comp. **63** (1994), 351–359.

[9] ———, *Some lower and upper complexity bounds for generalized Fourier transforms and their inverses*, SIAM J. Comput. **20**(3) (1991), 451–459.

[10] U. Baum, M. Clausen, and B. Tietz, *Improved upper complexity bounds for the discrete Fourier transform*, AAECC **2** (1991), 35–43.

[11] T. Beth, *On the computational complexity of the general discrete Fourier transform*, Theoret. Comput. Sci. **51** (1987), 331–339.

[12] ———, *Verfahren der schnellen Fourier–Transformation*, Teubner Studienbücher, Stuttgart, 1984.

[13] L. Bluestein, *A linear filtering approach to the computation of the discrete Fourier transform*, IEEE Trans. **AU-18** (1970), 451-455.

[14] A. Borodin and I. Munro, *The computational complexity of algebraic and numeric problems*, Elsevier, New York, 1975.

[15] O. Bratelli, *Inductive limits of finite dimensional $C^*$-algebras*, Trans. Amer. Math. Soc. **171** (1972), 195–234.

[16] P. Bürgisser, M. Clausen, A. Shokrollahi, *Algebraic Comlexity Theory*, Springer-Verlag, Berlin, 1996.

[17] D. Calvetti, *A stochastic roundoff error analysis for the fast Fourier transform,* Math. Comput., **56**(194) (1991), 755–774.

[18] C. Chui, *An introduction to wavelets*, Academic Press, NY, 1992.

[19] M. Clausen and U. Baum, *Fast Fourier transforms*, Wissenschaftsverlag, Mannheim, 1993.

[20] ———, *Fast Fourier transforms for symmetric groups, theory and implementation*, Math. Comp. **61**(204) (1993), 833–847.

[21] M. Clausen, *Fast Fourier transforms for metabelian groups*, SIAM J. Comput. **18** (1989), 584–593.

[22] ———, *Fast generalized Fourier transforms*, Theoret. Comput. Sci. **67** (1989), 55–63.

[23] ———, *Beiträge zum Entwurf schneller Spektraltransformationen*, Habilitationsschrift, Fakultät für Informatik der Universität Karlsruhe (TH), 1988.

[24] J. W. Cooley, *The re-discovery of the fast Fourier transform algorithm*, Mikrochimica Acta **III** (1987), 33–45.

[25] ———, *How the FFT gained acceptance*, Proceedings of the ACM conference on the history of scientific and numeric computation, Princeton, NJ May 13–15, 1987, 133–140.

[26] J. W. Cooley and J. W. Tukey, *An algorithm for machine calculation of complex Fourier series*, Math. Comp. **19** (1965), 297–301.

[27] I. Daubechies, *Ten lectures on wavelets*, SIAM, Philadephia, PA, 1992.

[28] P. Diaconis, *A generalization of spectral analysis with applications to ranked data*, Ann. Stat. **17** (1989), 949–979.
[29] ———, *Group representations in probability and statistics*, IMS, Hayward, CA, 1988.
[30] ———, *Average running time of the fast Fourier transform*, J. Algorithms **1** (1980), 187–208.
[31] P. Diaconis and D. Rockmore, *Efficient computation of the Fourier transform on finite groups*, J. Amer. Math Soc. **3**(2) (1990), 297–332.
[32] ———, *Efficient computation of isotypic projections for the symmetric group*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. **11**, L. Finkelstein and W. Kantor (eds.), 1993, 87–104.
[33] J. R. Driscoll and D. Healy, *Computing Fourier transforms and convolutions on the 2-sphere*, (extended abstract) Proc. $34^{th}$ IEEE FOCS, (1989) 344–349; Adv. in Appl. Math., **15** (1994), 202–250.
[34] J. R. Driscoll, D. Healy, and D. Rockmore. *Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs*, SIAM J. Comput. (to appear).
[35] C. Dunkl, *A Krawtchouk polynomial addition theorem and wreath products of symmetric groups*, Indiana Univ. Math. J. **25** (1976), 335-358.
[36] A. Dutt and V. Rokhlin, *Fast Fourier transforms for nonequispaced data*, SIAM J. Sci. Statist. Comput., (to appear).
[37] ———, *Fast Fourier transforms for nonequispaced data, II* Appl. and Comp. Harmonic Analysis **2** (1995), 85–100.
[38] D. F. Elliott and K. R. Rao, *Fast transforms: algorithms, analyses, and applications*, Academic, New York, 1982.
[39] A. Fässler and E. Stiefel, *Group theoretical methods and their applications*. Birkhäuser, Boston, MA, 1992.
[40] G. Gaudrey and R. Pini, *Bernstein's theorem for compact, connected Lie groups*, Math. Proc. Cambridge Phil. Soc. **99** (1986), 297–305.
[41] C. F. Gauss, *Theoria interpolationis methodo nova tractata*, Gauss' Collected Works Vol. 3, 1886, 265–303.
[42] I. M. Gel'fand and M. Graev, *Finite dimensional irreducible representations of unitary and full linear groups and related special functions*, Izv. Akad. Nauk SSSR, Ser. Math. **29** (1965), 1329–1356 (Russian).
[43] I. M. Gel'fand and M. Tsetlin, *Finite dimensional representations of the group of unimodular matrices*, Dokl. Akad. Nauk SSSR **71** (1950), 825–828 (Russian).
[44] F. Goodman, P. de la Harpe, and V. F. R. Jones, *Coxeter graphs and towers of algebras*, Springer-Verlag, New York, 1989.
[45] T. Hagedorn. *Multiplicities in restricted representations*. Ph.D. Thesis, Department of Mathematics, Harvard University, 1994.
[46] D. Healy and P. Kim, An empirical Bayes approach to directional data and efficient computation on the sphere, *Ann. Stat.*, (to appear).
[47] ——— *Spherical deconvolution with application to geometric quality assurance*, Technical Report, Department of Mathematics and Computer Science, Dartmouth College, 1993.
[48] D. Healy, D. Maslen, S. Moore, D. Rockmore, and M. Taylor, *Applications of a fast convolution algorithm on the 2-sphere*, (in preparation).
[49] D. Healy, S. Moore, and D. Rockmore, *Efficiency and reliability issues in a fast Fourier transform on the 2-sphere*, Technical Report, Department of Mathematics and Computer Science, Dartmouth College, 1994.
[50] ———, *Parallelizability of an FFT on the 2-sphere,* Technical Report, Department of Mathematics and Computer Science, Dartmouth College, 1994.
[51] M. T. Heideman, D. H. Johnson, and C. S. Burrus, *Gauss and the history of the fast Fourier transform*, Archive for History of Exact Sciences **34**(3) (1985), 265–277.
[52] E. Hewitt and K. Ross, *Abstract harmonic analysis*, vol. II, Die Grundlehren der mathematischen Wissenschaften **152**, Springer-Verlag, Berlin, 1963.
[53] R. Holmes, *Signal processing on finite groups*, Technical Report 873, MIT, Lincoln Laboratory, 1990.
[54] ———, *Mathematical foundations of signal processing, II*, Technical Report 781, MIT, Lincoln Laboratory, 1987.
[55] G. D. James, *The representation theory of the symmetric groups*, Lecture Notes in Math., vol. 682, Springer-Verlag, New York, 1978.
[56] K. Kanatani, *Group-theoretical methods in image understanding,* Springer-Verlag, NY (1990).

[57] M. Karpovsky and E. Trachtenberg, *Filtering in a communication channel by Fourier transforms over finite groups*, in Spectral Techniques and Fault Detection, M. Karpovsky (ed.), Academic Press, NY (1985), 179–212.
[58] A. Klimyk, *Matrix elements and Clebsch-Gordan coefficients of group representations*, Naukova Dumka, Kiev, 1979 (Russian).
[59] A. Klimyk and N. Vilenkin, *Relations between spherical functions of compact groups*, J. Math. Phys., **30** (6), June (1989), pp. 1219-1225.
[60] ———*Representations of Lie groups and special functions.* I, II, and III, Mathematics and Its Applications (Soviet Series), vols. 72, 74 and 75, Kluwer, Boston, 1991 and 1993.
[61] E. Lambina, *Matrix elements of irreducible representations of the group $K_n$ of orthogonal matrices*, Dokl. Akad. Nauk Byelorussian SSR **9** (1965), 77–81 (Russian).
[62] J. Lafferty and D. Rockmore, *Fast Fourier analysis for $SL_2$ over a finite field and related numerical experiments*, J. Exp. Math. **1** (1992) 115–139.
[63] S. Linton, G. Michler, and J. Olsson, *Fourier transforms with respect to monomial representations*, Math. Ann. **297** (1993), 253–268.
[64] D. Maslen, *Efficient computation of Fourier transforms on compact groups*, Max-Planck preprint MPI/95-8 (1995).
[65] ———*Fast Transforms and Sampling for Compact Groups*, Ph.D. Thesis, Department of Mathematics, Harvard University, 1993.
[66] ———*Sampling of functions and sections for compact groups*, preprint.
[67] ———, *A polynomial approach to orthogonal polynomial transforms*, Max-Planck preprint MPI/95-9 (1995).
[68] D. Maslen and D. Rockmore, *Separation of variables and the efficient computation of Fourier transforms on finite groups, I*, submitted for publication.
[69] ———, *Separation of variables and the efficient computation of Fourier transforms on finite groups, II*, in preparation.
[70] ———, *Adapted diameters and the efficient computation of Fourier transforms of finite groups*, in Proceedings of the 1995 ACM-SIAM Symposium on Discrete Algorithms (to appear).
[71] H. Möller, *On the construction of cubature formulae with few nodes using Groebner bases*, in Nato Adv. Sci. Inst. Ser. C **203** (1987), P. Keast, G. Fairweather (eds.), Reidel, Dordrecht, 177-192.
[72] S. Moore, *Efficient stabilization methods for fast polynomial transforms*, Ph.D. Thesis, Dartmouth College, NH, 1994.
[73] S. Moore, D. Healy, and D. Rockmore. *Symmetry stabilization for polynomial evaluation and interpolation*, Linear Algebra Appl. **192** (1993), 249–299.
[74] A. Oppenheim and R. Schafer, *Discrete-time signal processing.* Prentice Hall, NJ, 1989.
[75] S. Orszag, *Fast eigenfunction transforms*, in Science and Computers, G. C. Rota (ed.), (A Volume Dedicated to Nicholas Metropolis), Adv. Math. Suppl. Stud., vol. 10, Academic Press, Orlando, FL, 1984, 22–30.
[76] R. Pini, *Bernstein's Theorem on $SU(2)$*, Boll. Un. Mat. Ital. A (6) **4**-A (1985), 381-389.
[77] L. Rabiner, R. Schafer, and C. Rader, *The chirp-z transform and its applications*, Bell System Tech. J. **48** (1969), 1249-1292.
[78] C. Rader, *Discrete Fourier transforms when the number of data samples is prime*, IEEE Proc. **56** (1968), 1107-1108.
[79] G. U. Ramos, *Roundoff error analysis of the fast Fourier transform*, Math. Comp. **25** (1971), 757–768.
[80] D. Rockmore, *Applications of generalized FFTs*, this volume.
[81] ———, *Fast Fourier transforms for wreath products*, Appl. Comput. Harmon. Anal. **4** (1995), 34–55.
[82] ———*Efficient computation of Fourier inversion for finite groups*, J. Assoc. Comput. Mach. **41**(1) (1994), 31–66.
[83] ———*Fast Fourier analysis for abelian group extensions*, Adv. in Appl. Math. **11** (1990), 164–204.
[84] P. Schwartztrauber, *The vector harmonic transform method for solving partial differential equations in spherical geometry*, Mon. Wea. Rev. **121**(12) (1993), 3415-3437.
[85] D. Stanton, *Orthogonal polynomials and Chevalley groups*, in Special Functions: Group Theoretical Aspects and Applications, R. Askey et al. (eds.), Kluwer, Dordrecht, 1984, 87-128.

[86] G. Steidl and M. Tasche, *A polynomial approach to fast algorithms for discrete Fourier-cosine and Fourier-sine transforms*, Math. Comp. **56** (1991), 281–296.
[87] L. Stiller, *Exploiting symmetry on parallel architectures*, Ph. D. Dissertation, Department of Computer Science, The Johns Hopkins University, Baltimore, MD, 1995.
[88] E. Thoma, *Die Einschränkung der Charaktere von* $GL(n,q)$ *auf* $GL(n-1,q)$, Math. Z., **119** (1971) 321–338.
[89] R. Tolimieri, M. An, and C. Lu, *Algorithms for discrete Fourier transform and convolution*, Springer-Verlag, New York, 1989.
[90] C. Van Loan, *Computational framework for the fast Fourier transform*, SIAM, Philadelphia, 1992.
[91] N. Vilenkin, *Special functions and the theory of group representations*, Translations of Mathematical Monographs, **22**, A.M.S., Providence RI, 1968.
[92] A. S. Willsky, *On the algebraic structure of certain partially observable finite-state Markov processes*, Inform. Contr. **38**, 179–212 (1978).
[93] F. Yates, *The design and analysis of factorial experiments*, Imp. Bur. Soil Sci. Tech. Comm. **35** (1937).
[94] A. Zakhor, R. Weisskoff, and R. Rzedzian, *Optimal sampling and reconstruction of MRI signals resulting from sinusoidal gradients*, IEEE Trans. Signal Process. **39** (1991) 2056–2065.
[95] D. Želobenko, *Compact Lie groups and their representations*, Translations of Mathematical Monographs, **40**, A.M.S., Providence RI, 1973.

Department of Mathematics, Universiteit Utrecht, 3584 CD, Utrecht, Netherlands
*E-mail address*: maslen@math.ruu.nl

Department of Mathematics, Dartmouth College, Hanover, NH 03755
*Current address*: School of Mathematics, Institute for Advanced Study, Princeton, NJ 08544
*E-mail address*: rockmore@cs.dartmouth.edu